
MARZO 1986
Nº 12 ₳ 2
REP. ARGENTINA
K64
COMPUTACION PARA TODOS
La Odisea de la
Quinta Generación
13 Programas
Inéditos
Concurso:
El Programador
del Año
Suplemento de
Aplicaciones para C64,
CZ 1000/2000, TK 83/90X:
Alarma Anti-robo
Interfase de Control
Procesador de Textos
Sintetizador de Voz



SUMARIO

QUINTA GENERACION

Los japoneses esperan desarrollar máquinas que piensen como los hombres. La polémica que se generó es analizada por Edward Feigenbaum y Pamela McCorduck.

PAG. 8

ALARMA ANTI-ROBO

Ofrecemos un proyecto con un pequeño programa, para utilizar la Drean Commodore 64 como una alarma, con diversas aplicaciones. Para su instrumentación utilizaremos el USER PORT, el cual es fácilmente accesible en la parte posterior de la C-64.

PAG. 27

INTERFASE DE CONTROL

Las máquinas diseñadas en base al microprocesador Z-80 disponen de buenos controladores de periféricos. Uno de ellos, el Z-80 PIO puede ser utilizado para el desarrollo que se explica en estas páginas.

PAG. 34

SINTETIZADOR DE VOZ

Incluímos programas que permiten que nuestra TS 2068, Spectrum o TK 90X "hable" con un vocabulario ilimitado, eliminando las dificultades que puede acarrear el almacenamiento de sonidos.

PAG. 37

CARTA DEL DIRECTOR

Las profecías de los autores de ciencia ficción vuelven a hacerse realidad, superando incluso lo imaginado. Cuando hace dos décadas Arthur Clark en su "2001: una odisea espacial" describía a la supercomputadora HAL 9000, seguramente no pensaba que las máquinas inteligentes estarían listas diez años antes de la fecha que él había soñado. Y éso es lo que sucederá, con el avance de la quinta generación, de la cual hablamos en este número.

Hoy ya se sabe que la informática no sólo sirve para manejar vuelos en el cosmos, sino que también los pequeños equipos tienen múltiples aplicaciones, y algunos ejemplos damos en nuestro suplemento especial en esta edición.

De manera que las máquinas estén al servicio de los hombres, y no a la inversa (como alertó el escritor norteamericano con su HAL 9000)

CRISTIAN PUSSO

PROGRAMAS INEDITOS

K64
COMPUTACION PARA TODOS

AÑO 1 Nº 12 MARZO DE 1986

Director General
Ernesto del Castillo
Director Editorial
Cristian Pusso
Director Periodístico
Fernando Flores

Director Financiero
Javier Campos Malbrán
Coordinador
M.G. Verdomar Weiss
Secretaria
Moni Ocampo

Diagramación
Fernando Amengual
Departamento de Avisos
Oscar Devoto

Departamento de Publicidad
Jefe: Dolores Urien
Promotora:
Mónica Garibaldi

K-64 es una Revista mensual editada por Editorial PROEDI S.A. (e./f.), Cerrito 1320, 1º Piso, Buenos Aires, Te.: 42-9681/9. Registro Nacional de la Propiedad Intelectual: 313.837 M. registrada. Queda hecho el depósito que indica la Ley 11.723 de Propiedad Intelectual.
Impresión: Calcotam. Fotocromo tapa: Columbia. Fotocomposición: Van Waveren.-
Distribuidor en Capital: MARTINO, Juan de Garay 358, P.B. Capital. T.E. 361-6962 Distribuidor interior: DGP, Hipólito Yrigoyen 1450, Capital T.E. 38-9266/9800. Los ejemplares atrasados se venderán al precio del último número en circulación.

CORREO ARGENTINO CASA CENTRAL | FRANQUEO PAGO CONCESION Nº 2538 | TARIFA REDUCIDA CONCESION Nº 836

Empresario francés en Argentina


Georges Pebereau

Visitó nuestro país Georges Pebereau, director general del grupo CGE, el primer holding empresario de Francia en el sector electrónico y de telecomunicaciones, que a su vez se ubica dentro de las primeras cinco grandes en el mundo dentro de la especialidad. Entre sus principales áreas se destacan las telecomunicaciones. la informática, la telemática, robótica y el desarrollo de la inteligencia artificial.
Dentro de la actividad que George Pebereau desarrolló en Argentina, se entrevistó con el presidente Raúl Alfonsín, con motivo de hacer participar a Alcatel Thomson, una empresa del holding francés que él dirige, en el concurso para la fabricación de equipos de computación y transmisión numérica. La propuesta presentada por Alcatel Thomson apunta a 120 mil líneas por año, que requerirán una inversión de cien millones de francos.
En cuanto a los componentes, se procurará establecer una relación de cooperación en el campo de la microelectrónica con los laboratorios y las industrias argentinas.
Cabe destacar que Alcatel Thomson es líder mundial en conmutación pública, y, las instalaciones que hasta hoy realizó representan un 30 por ciento del mercado mundial.

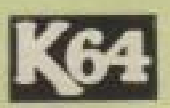


Actividad de la Subsecretaría de Informática y Desarrollo

El doctor Carlos María Correa, subsecretario de Informática, dio a conocer que "se han iniciado estudios sobre las nuevas tendencias legislativas en materia de semiconductores de alta integración (chips)". Mencionando el reciente dictado de leyes que establecen un derecho especial de propiedad en EEUU y Japón, señaló que "los países que buscan ingresar al campo. de la informática deben estudiar el impacto de estas legislaciones, así como el convenio internacional sobre la materia que ya ha sido elaborado y propuesto por la Organización Mundial de la Propiedad Intelectual". Adelantó también que "un representante de nuestro país asistirá el próximo mes a una reunión internacional sobre el tema, y procurará sentar posiciones comunes con otros países en desarrollo, y en particular los latinoamericanos".

La informática y los derechos indivuales

Por otro lado, el doctor Correa comentó el avance registrado en los trabajos de la misión que analiza el impacto de la informática sobre la privacidad y los derechos individuales. "La manipulación de datos de las personas —recordó— pone en riesgo no sólo la confidencialidad de información personal, sino que genera indefensión frente a datos cuya exactitud no se puede corroborar y corregir". Dentro del trabajo realizado, el funcionario comentó que la comisión "realizó un análisis de la legislación comparada, en



particular de las leyes austríaca, danesa, francesa y noruega, e identificó doce principios generales".
Algunos de los principios son, de la justificación social; de la limitación de la recolección; de la calidad de la información; de la especificación del propósito o la finalidad; de la confiabilidad; de salvaguarda de la seguridad; de política o apertura; de la limitación en el tiempo; de control y de la participación individual.
Esta comisión que lleva adelante estos estudios está integrada por los doctores Carlos Nino, Ricardo Guibourg, Pedro Molinero y Eduardo Novoa Monreal.

Publicación de NCR Argentina

La empresa NCR ha lanzado el primer número de Presencia, publicación institucional de la empresa en Argentina. Esta nueva publicación, que se editará trimestralmente, se distribuye a consultores y a clientes actuales y potenciales. En ella se encontrarán testimonios de usuarios de equipos NCR, información sobre sistemas que se presenten en el mercado y artículos sobre novedades de la empresa.

Combaten la piratería

Dos nuevos recursos para combatir la piratería de software están prontos a ser utilizados. El primero de ellos consiste en una lente especial que se suministra junto con el soft, con el cual hay que leer un código que aparece en la pantalla. Este sistema permite realizar copias "back-up" de seguridad para el propietario del soft.
El otro sistema resultará útil también para los comercios expendedores de soft. Consiste en programas en "cartridges" del tamaño de una tarjeta de crédito que se graban en el momento de venderse. Para usarlos, se necesitará de una interface especial de bajo costo, adaptable a las marcas más populares.
Entre otras ventajas, los programmas se cargarán inmediatamente, ocuparán poco espacio físico y no correrán peligro de borrarse.

Recibieron sus premios

La foto registra el momento en que Jorge Mariaschin y Jorge Rossi concurrieron a nuestras oficinas para retirar los premios que obtuvieron como ganadores del tercer concurso K-64.
El primero recibió los dos pasajes a Bariloche, ida y vuelta en avión, mientras que el segundo se llevó



Jorge Rossi (segundo premio), y Jorge Mariaschin (primer premio).

los dos boletos para Punta del Este, también ida y vuelta en avión. Como se recordará los otros galardonados ganaron una impresora Alphacom 32, un grabador para computadora y una mesa especial

para informática. ¿Quién se adjudicará el lingote de oro de la nueva selección K-64 "El programador del año '86"? (ver bases en este número).

Sistema/36 PC de IBM

IBM anunció el lanzamiento en el mercado argentino del Sistema/36 PC (IBM 5364) que debido a su gran variedad de aplicaciones, es un equipo apto para servicios profesionales, comerciales y como procesador de datos para la pequeña y mediana empresa.
Es de fácil aplicación y de costo accesible. Su tamaño permite que pueda colocarse sobre un escritorio. En cuanto a sus particularidades técnicas podemos mencionar que soporta un máximo de cuatro estaciones de trabajo conectadas localmente, la primera de las cuales debe ser una PC, y hasta 64 estaciones adicionales remotas conectadas por líneas de comunicación. Por otra parte, utiliza un disco fijo desarrollado y fabricado por IBM con una capacidad de almacenamiento de 40 millones de caracteres, que



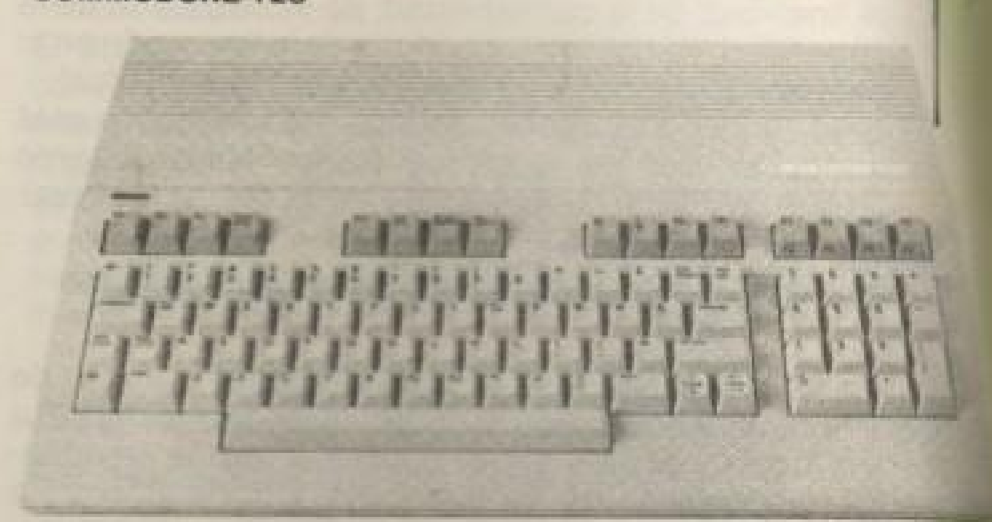
Nuevo componente del S/36 PC

se puede ampliar a 80 millones de caracteres en una segunda unidad. Además cuenta con una unidad de diskette que puede leer y grabar diskettes de 1,2 Megabytes de capacidad.
Una importante ventaja del sistema Sistema/36 PC es la posibilidad de acceder a todo el software que hay disponible en el mercado local para Sistema/36.

Escuela Superior Latinoamericana de Informática

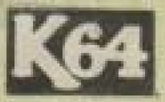
La Escuela Superior Latinoamericana de Informática (ESLAI), creada en 1985 bajo el auspicio del gobierno argentino y de organismos internacionales especializados, inició en octubre de ese mismo año sus actividades regulares, dirigidas a estimular la formación de alto nivel de docentes e investigadores latinoamericanos en informática.

COMMODORE 128



C128 en Argentina

Es posible encontrar, en nuestro mercado, la nueva COMMODORE 128 PERSONAL COMPUTER. Esta poderosísima computadora puede definirse como "tres en una" ya que puede trabajar en modo C-64 permitiendo que todos los programas escritos para él puedan correrse en la 128, en el modo C-128 con el nuevo BASIC 7.0 y en modo CP/M trabajando con el microprocesador Z80. En este último modo se puede correr los cientos de programas comerciales disponibles para computadoras de 8 bytes con CP/M como sistema operativo.
Además puede leer ocho formatos de CP/M incluyendo aquellos para OSBORNE, EPSON e IBM CP/M 86.
Commodore lanzó en EE.UU. junto con la PC el nuevo Disk Drive 1571 que presente, también, tres modos de operación. Cuando la C-128 trabaja en modo 64, el 1571 se comporta como una 1541 cien por cien. Respeta la baja velocidad de transferencia de información de ésta (300 cps) y su capacidad de almacenamiento (170 Kbytes). En modo 128, la velocidad de transferencia aumenta a 1500 cps (5 veces más rápido que la 1541) y la capacidad se eleva a 350 Kbytes.
Finalmente, en el modo CPM, la velocidad de transferencia llega a los 3500 cps (de 10 a 12 veces más rápido que la 1541) con cerca de 410K de almacenamiento de datos.
Algunas de sus características son:
Microprocesador:
Z80 (modo CP/M)
Memoria:
128K RAM (expandible a 512K)
Clock:
4 MHz (modo CP/M)
1 o 2 MHz (modo 64)
Almacenamiento en disco (también trabaja en cassette):
350-410K: 128 / CP/M
170K: 64
Sonido:
3 voces con generador de ruidos
Teclado: 92 teclas; 14 numéricas; 8 de función.
Pantalla:
40/80 columnas
Otro detalle importante es que puede trabajar en alta resolución directamente.







Buenos Aires, Sheraton Hotel, del 19 al 23 de mayo de 1986

El COMITE ACADEMICO DEL CONGRESO "USUARIA '86", invita a la presentación de trabajos técnicos, para ser expuestos en las siguientes áreas temáticas:

- Seminario de Banca e Informática.
- Seminario de Productividad Industrial e Informática.
- Seminario de Educación e Informática.
- Seminario de Gobierno e Informática.
- Seminario de Informática y Teleinformática.

Los trabajos serán recibidos hasta el 1º de abril de 1986, en la "Secretaría del Comité Académico - USUARIA '86"

Asociación Argentina de Usuarios de la Informática - USUARIA
Hipólito Yrigoyen 1427, Piso 8, (1089) Buenos Aires, Argentina
Teléfonos: 38-6579 / 38-7906.

Auspician: • Subsecretaría de Informática y Desarrollo. • Oficina Intergubernamental para la Informática (IBI). • Oficina Regional de Ciencia y Técnica para América Latina y el Caribe (UNESCO). • Federación Latinoamericana de Usuarios de la Informática (FLAI). • Centro Latinoamericano de Matemática e Informática (CONICET-UNESCO).

LA ODISEA DE LA QUINTA GENERACION

En menos de una década los japoneses esperan desarrollar máquinas superdotadas que piensen como los hombres. La promesa desató polémicas, que son analizadas en el libro "La Quinta Generación", de Edward Feigenbaum y Pamela McCorduck, reeditado por Sudamericana-Planeta, del que publicamos algunos comentarios.

Una máquina tan inteligente como una persona

La dificultad con que la mayoría de nosotros se enfrenta cuando ha de pensar en máquinas inteligentes es que nuestro concepto de «máquina» está condicionado por las máquinas que nos han rodeado toda la vida. Su función en casi todos los casos es tratar la energía: es decir, ampliar, distribuir, transformar o modificar de algún modo la energía. Así, por ejemplo, el automóvil transforma la energía de un combustible fósil (a su vez transformado ya por el refinamiento) en energía cinética, y esta transformación amplifica la energía cinética humana y satisface objetivos humanos. En coche podemos ir más lejos que andando. Además todas estas transformaciones de la energía pueden describirse claramente mediante las disciplinas científicas clásicas.

Sin embargo, el ordenador es un tipo diferente de máquina. No trata energía, sino información. Como es lógico, entra alguna energía en el proceso, del mismo modo que la transformación de la información interviene también en los sistemas telefónicos o de radiodifusión, pero las transformaciones de energía en el ordenador son sus rasgos menos interesantes, aunque puedan serlo para algunos ingenieros. Para comprender la función esencial de los ordenadores —en su calidad de máquinas— debemos quitarnos de la cabeza las viejas metáforas y empezar a pensar de un modo nuevo. El ordenador es el aparato principal de la era de la información. Su objetivo es sin duda tratar la información: transformarla, amplificarla, distribuirla y modificarla en general. Pero hay algo más importante, y es que el ordenador produce información. La esencia de la revolución informática es que la carga que supone producir el saber futuro del mundo recaerá sobre los aparatos mecánicos y no sobre las cabezas humanas. Aunque el Eclesiastés diga lo contrario, quizá haya algo nuevo bajo el sol.

Con todo, estos aparatos no han recibido el nombre adecuado, y esto puede confundirnos. La palabra computadora u ordenador, con sus reminiscencias de cálculos y contaje, sólo nos informa sobre la utilización histórica de la máquina, no sobre su potencial. Los japoneses se han dado cuenta de ello y han bautizado a su Quinta Generación de ordenadores con el nombre de máquinas para «el tratamiento informático del saber», lo cual sugiere a su vez que el saber y la información son dos entidades distintas.

Nosotros hemos vivido una época de transición, con teléfonos y televisores que están a caballo de ambos mundos, el de la información y el de la energía. La llegada de la primera generación de ordenadores nos introdujo claramente en la nueva era. Y ahora estamos avanzando hacia la fase siguiente: la era de las Máquinas Inteligentes.

Sentimos en nuestra nuca el cálido aliento de los maniáticos: «¿Qué significa esto de inteligente? Estas máquinas pretendidamente inteligentes, ¿no lo serán tanto como una persona, supongo? Es imposible que lo sean, porque todo lo que saben se lo enseñan las personas.»

«Piensa —dice Feigenbaum a McCorduck un día— que no existe ninguna máquina que sea tan inteligente como una persona.»

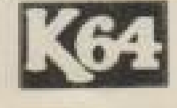
Ella le mira sorprendida. ¿Son una estafa todos estos programas que generan la capacidad de los especialistas? ¿Quizás ella no ha captado bien sus palabras, y le pide que lo repita; pero tampoco así acaba de entenderlo.

«¿Podrías explicármelo mejor?.»

«Muy fácilmente: se empieza escogiendo una tarea para que una máquina la lleve a cabo. Se detalla esta tarea muy precisamente, basándose en el saber humano. Se utilizan para ello los conocimientos del equipo de especialistas, pero la máquina continúa siendo menos inteligente que ellos. Sin embargo, el hecho de tener el programa y los conocimientos expuestos detalladamente permite descubrir de modo inmediato la manera de introducir mejoras. Y de repente el programa supera al hombre. Resulta imposible fijar el momento preciso en que la máquina es exactamente tan inteligente como una persona. Durante un tiempo no era tan inteligente y de pronto se convierte en más inteligente.

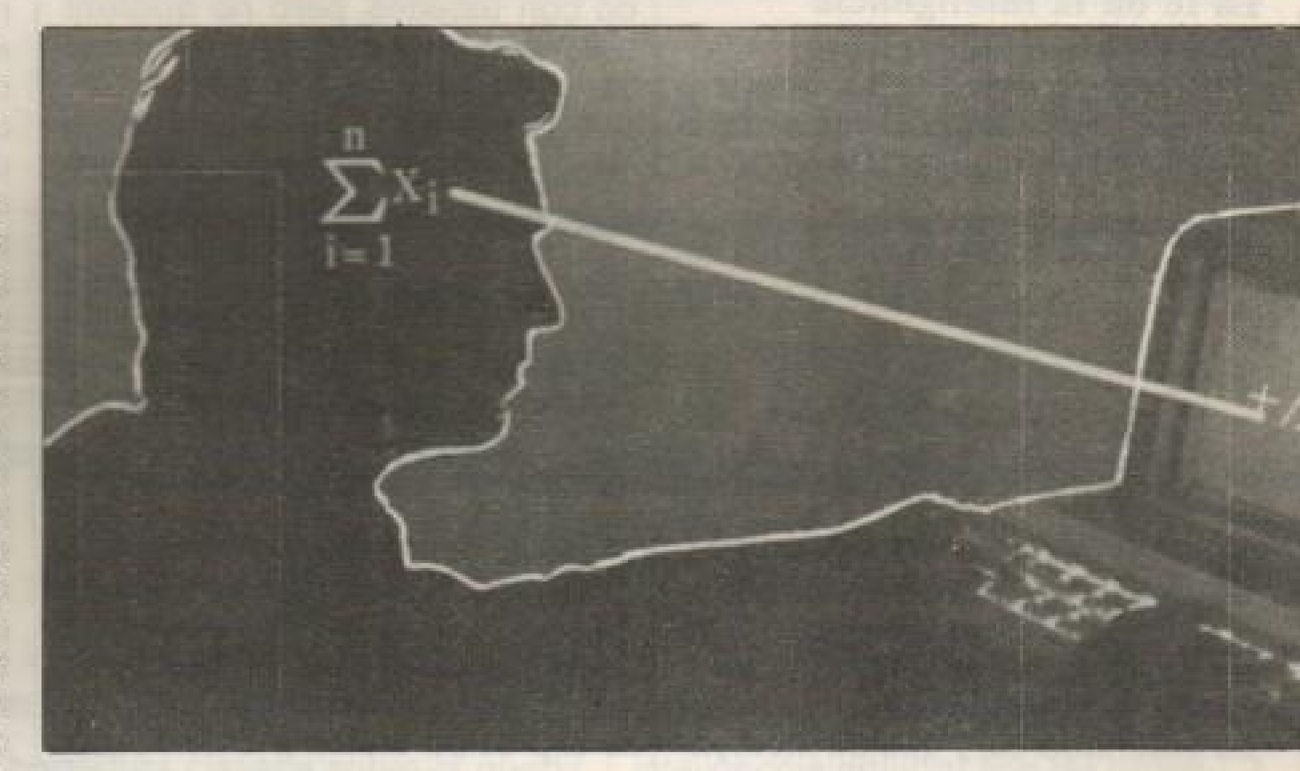
«Las máquinas cuidan metódicamente del detalle, son incapaces de cansarse, son inmunes al aburrimiento y tienen una elevadísima velocidad. Todo ello, unido a su poder de raciocinio y a su información, está empezando a producir

"Debemos estar orgullosos de reconocer nuestras limitaciones y de inventar una tecnología que pueda compensarlas".



un saber que a menudo es más rápido y mejor —más «inteligente»— que el de las personas que los enseñaron.

»Y debemos preguntarnos con toda humildad: ¿Qué inteligentes eran las personas que enseñaron a estas máquinas? Dentro de la escala temporal de la evolución, los animales pensantes son unos recién llegados. La evolución no dispuso de mucho tiempo para dedicarse a perfeccionar la capacidad de conocimiento del hombre. Sin duda las respuestas correctas a las preguntas: "¿Qué complejo patológico está afectando a mi paciente?", o "¿Qué plan experimental permitirá obtener un clon de un gen particular?", o "¿Cómo puedo sintetizar el nuevo fármaco que acabo de descubrir?" están debajo de nuestras mismas narices, pero somos incapaces de verlas. Sin embargo, existen ya programas de sistemas especializados, claramente primitivos, que pueden responder a tales preguntas. En el futuro, máquinas más inteligentes responderán de modo rutinario a preguntas más difíciles.



»Las personas sabemos muy bien convertir las señales sensoriales en símbolos de conocimiento y resolver problemas que exigen sentido común. Pero cuando nos enfrentamos con grandes cantidades de datos nos acobardamos: nos mostramos poco sistemáticos y olvidadizos, nos aburrimos, nos distraemos. La escritura y la tecnología del libro nos ayudaron a superar algunos de estos problemas, los ordenadores inteligentes e interactivos nos ayudarán algo más. Debemos sentirmos orgullosos de poder reconocer nuestras limitaciones y de inventar una tecnología que puede compensarlas.»

NOTA: IA (Inteligencia Artificial)

La fe en la inteligencia artificial

Un participante en la conferencia de la Quinta Generación se levantó para formular algunas objeciones a lo que había oído. Las objeciones no eran serias, pero el participante concluyó así: »Creo, resumiendo mi punto de vista, que estamos interesados en la siguiente generación de ordenadores, y que de momento suponemos que estos ordenadores serán máquinas de IA. En términos generales estoy de acuerdo con ustedes, pero no quisiera perder de vista que algunas personas asistentes no creen en la IA y que, por tanto, desearían considerar a los ordenadores de la Quinta Generación como algo diferente.» La expresión era curiosa: «no creen en la IA», como si la IA fuera cuestión de fe mística, como si fuera algo no sujeto a prueba empírica. De hecho la expresión aludía a una controversia de trascendencia mucho mayor que la referente a la elección del mejor lenguaje de programación posible o a la idoneidad del enfoque basado en el saber para conseguir que los ordenadores se comportaran de modo inteligente, o a cualquier otra de las disputas científicas que han animado el campo de la investigación de la inteligencia artificial en su más de un cuarto de siglo de existencia.

"Producir el saber futuro del mundo recaerá sobre los aparatos mecánicos y no sobre las cabezas humanas".

Decidir si la Quinta Generación sería una máquina de razonamiento simbólico con un diseño nuevo o más bien una versión mayor y mejor de las cuatro primeras generaciones de ordenadores era algo que se resolvería a su debido tiempo.

Lo que no se resolvería, por lo menos en las mentes de quienes lo dudaban de entrada, era la posibilidad de creer en la inteligencia artificial. Decir que uno no cree en la inteligencia artificial —y hay mu-

EL ARTE DE CONSTRUIR UN CEREBRO

Hace menos de dos décadas, Arthur C. Clarke, en "2001, una odisea espacial" (reeditado ahora por Hyspamérica), anticipaba el nacimiento de las extrañas criaturas tecnológicas.

Al sexto miembro de la tripulación no le importaban nada todas esas cosas, pues no era humano.

Era el sumamente perfeccionado computador HAL 9.000, cerebro y sistema nervioso de la nave.

HAL (sigla de Computador ALgorítmico Heurísticamente programado, nada menos) era una obra maestra de la tercera promoción de computadores. Ello parecía ocurrir a intervalos de veinte años, y mucha gente pensaba ya que otra nueva creación era inminente.

La primera había acontecido en 1940 y pico, cuando la válvula de vacío hacía tiempo anticuada, había hecho posible tan toscos cachivaches de alta velocidad como ENIAC y sus sucesores. Luego, en los años sesenta habían sido perfeccionados sólidos ingenios microelectrónicos. Con su advenimiento, resultaba claro que inteligencias artificiales cuando menos tan poderosas como la del Hombre, no necesitaban ser mayores por mesas de despacho... caso de que se supiera cómo construirlas.

Probablemente nadie lo sabría nunca; mas ello no importaba. En los años ochenta, Minsky y Good habían mostrado cómo podían ser generadas automáticamente redes nerviosas autorreplicadas, de acuerdo con cualquier arbitrario programa de enseñanza. Podían construirse cerebros artificiales mediante un proceso asombrosamente análogo al desarrollo de un cerebro humano. En cualquier caso dado, jamás se sabrían los detalles precisos, y hasta si lo fueran, serían millones de veces demasiado complejos para la comprensión humana.

Sea como fuere, el resultado final fue una máquina-inteligencia que podía reproducir —algunos filósofos preferían aún emplear la palabra "remedar"— la mayoría de las actividades del cerebro humano, y con mucha mayor velocidad y seguridad. Era sumamente costosa, y sólo habían sido construidas hasta la fecha unas cuantas unidades de la HAL 9.000; pero estaba ya comenzado a sonar un tanto a hueca la vieja chanza de que siempre sería más fácil hacer cerebros orgánicos mediante un inhábil trabajo.

Hal había sido entrenado para aquella misión tan esmeradamente como sus colegas humanos... y a un grado de potencia mucho mayor, pues además de su velocidad intrínseca, no dormía nunca. Su primera tarea era mantener en su punto los sistemas de subsistencia, comprobando continuamente la presión del oxígeno, la temperatura, el ajuste del casco, la radiación y todos los demás factores inherentes de los que dependían las vidas del frágil cargamento humano. Podía efectuar las intrincadas correcciones de navegación, y ejecutar las necesarias maniobras de vuelo cuando era el momento de cambiar de rumbo. Y podía atender a los hibernadores, verificando cualquier ajuste necesario a su ambiente, y distribuyendo las minúsculas cantidades de fluidos intravenosos que los mantenían con vida.

Las primeras generaciones de computadoras habían recibido la fuerza necesaria a través de teclados de máquinas de escribir aumentados, y

chas personas que lo afirman añadiendo todo tipo en énfasis, razones e irritación— equivale a decir que uno no cree que de una máquina pueda decirse que piense, haga lo que haga esta máquina.

Cuando alguien propuso por primera vez que quizás un ordenador llegaría a comportarse inteligentemente, se levantó un enérgico coro de protestas. Por inteligente que fuera el comportamiento de los sucesivos ordenadores, los incrédulos no se dejarían convencer. La misma frase de «creer en» sugería el dogma, la doctrina, o el poder o no poder según los catecismos. *«Moi, je suis socialiste»*, dice a Stephen Daedalus uno de sus amigos. *«Je ne crois pas en l'existence de Dieu.»* Por mi parte yo trabajo con componentes y circuitos. Yo no creo en la existencia de la inteligencia artificial.

Feigenbaum había escuchado este argumento tan a menudo que ahora podía contar una pequeña historia sobre el tema. La protagonizaba el gran físico Niels Bohr, a quien visitaba uno de los jóvenes físicos de Europa. El joven científico se escandalizó cuando vio una herradura clavada sobre la puerta del gran físico. «Profesor Bohr —le dijo—, ¿supongo que usted no cree en esta vieja superstición?» El físico reflexionó un momento y luego respondió alegremente: «Dicen que surte efecto tanto si uno cree en ella como si no.»

Una escafandra para la mente

Una de las objeciones que los científicos vecinos formulan contra la inteligencia artificial son las profecías, en su opinión disparatadas e incluso irresponsables, hechas por quienes trabajan en este campo.

“Máquinas más inteligentes responderán de modo rutinario a preguntas cada vez más difíciles”.

habían replicado a través de impresores de alta velocidad y despliegues visuales. Hal podía hacerlo también así, de ser necesario, pero la mayoría de sus comunicaciones con sus camaradas de navegación se hacían mediante la palabra hablada. Poole y Bowman podían hablar a Hal como si fuese un ser humano, y él replicaría en el perfecto y más puro inglés que había aprendido durante las fugaces semanas de su electrónica infancia.

Sobre si Hal pudiera realmente pensar, era una cuestión que había sido establecida por el matemático inglés Alan Turing en los años cuarenta. Turing había señalado que, si se podía llevar a cabo una prolongada conversación con una máquina —indistintamente mediante máquina de escribir o micrófono— sin ser capaz de distinguir entre sus respuestas y las que pudiera dar un hombre, en tal caso la máquina estaba pensando, por cualquier sensible definición de la palabra. Hal podía pasar con facilidad el test de Turing.

Y hasta podía llegar el día en que Hal tomase el mando de la nave. En caso de emergencia, si nadie respondía a sus señales, intentaría despertar a los durmientes miembros de la tripulación, mediante una estimulación eléctrica y química. Y si no respondían, pediría nuevas órdenes por radio a la Tierra.

Y entonces, si tampoco la Tierra respondiese, adoptaría las medidas que juzgara necesarias para salvaguardia de la nave y la continuación de la misión... cuyo real propósito sólo él conocía, y que sus colegas humanos jamás habrían sospechado.

Poole y Bowman se habían referido a menudo humorísticamente a sí mismos como celadores o conserjes a bordo de una nave que podía realmente andar por sí misma. Se hubiesen asombrado mucho, y su indignación hubiera sido más que regular, al descubrir cuánta verdad contenía su chanza.

....

Había otros pensadores —Bowman lo hallaba así también— que sustentaban puntos de vista aún más avanzados. No creían que seres realmente evolucionados poseyeran en absoluto un cuerpo orgánico. Más pronto o más tarde, al progresar su conocimiento científico, se desembarazarían de la morada, propensa a las dolencias y a los accidentes, que la Naturaleza les había dado, y que los condenaban a una muerte inevitable. Reemplazarían su cuerpo natural a medida que se desgastasen —o quizás antes— con construcciones de metal o de plástico, logrando así la inmortalidad. El cerebro podría demorarse algo como último resto del cuerpo orgánico, dirigiendo sus miembros mecánicos y observando el Universo a través de sus sentidos electrónicos... sentidos mucho más finos y sutiles que aquellos que la ciega evolución pudiera desarrollar jamás.

Hasta en la Tierra se habían dado ya los primeros pasos en esa dirección. Había millones de hombres, que en otras épocas hubiesen sido condenados, que ahora vivían activos y felices gracias a miembros artificiales, riñones, pulmones y corazones. A este proceso sólo cabía una conclusión... por muy lejana que pudiera estar.

Y eventualmente, hasta el cerebro podría incluirse en él. No resultaba esencial como sede de la conciencia, como lo había probado el desarrollo de la inteligencia electrónica. El conflicto entre mente y máquina podía ser resuelto al fin en la tregua eterna de la completa simbiosis...

Más, ¿era aún esto el fin? Unos cuantos biólogos inclinados a la mística, iban todavía más lejos. Atando cabos en las creencias de diversas religiones, especulaban que la mente terminaría por liberarse de la materia. El cuerpo-robot, como el de carne y hueso, sería solamente un peldaño hacia algo que, hacía tiempo, habían llamado los hombres “espíritu”.

Y si más allá de esto había algo, su nombre sólo podía ser Dios.

Es cierto que se han hecho profecías que no se han cumplido todavía. Por ejemplo, un conjunto de predicciones en 1958 dijo que al cabo de diez años un ordenador sería el campeón mundial de ajedrez. Pasaron diez años y el cabo de este tiempo la mayoría de los científicos habían perdido interés en el ajedrez de ordenador. Pero pasados veinte años los ordenadores jugaban al ajedrez con una categoría suficiente para ganar en los campeonatos. Casi toda la investigación que había impulsado a estos ordenadores por encima del nivel de principiantes se había hecho con el equivalente en IA de los aficionados caseros. Las máquinas de ajedrez que actualmente funcionan a nivel de campeonato y que, por tanto, juegan al ajedrez mejor que el 99 por ciento de nosotros, se deben todavía al trabajo apasionado de un pequeño grupo de investigadores y no constituyen, como se había pensado al hacer aquella predicción, un campo central de pruebas donde descubrir reglas de comportamiento inteligente. Desde el punto de vista intelectual, un buen ajedrecista no es ni más ni menos que un buen ajedrecista. Como veremos más tarde, este resultado permitió hacer un descubrimiento importante sobre la inteligencia como especialización del saber.

Los especialistas de todos los campos no se cansan nunca de formular predicciones sobre el futuro. Si se hacen comparaciones, la inteligencia artificial se ha acercado mucho más al cumplimiento de sus profecías que muchas otras ramas de la ciencia. Es evidente que las predicciones cumplen funciones importantes de tipo psicológico, social y planificador, sea cual fuere su correspondencia con el futuro eventual. ¿Por qué, entonces, tanta gente se siente ofendida cuando se hacen predicciones sobre la inteligencia artificial?

La respuesta a esta pregunta parece evidente. El elemento ofensivo en las predicciones formuladas por los investigadores de inteligencia artificial es el mismo elemento que ofende a algunas personas cuando consideran la idea de la inteligencia artificial en sí misma: es el hecho de que exista. Está claro que los científicos están creando ya máquinas inteligentes cuyo objetivo es amplificar la inteligencia humana, una especie de escafandra autónoma, que permitirá a la mente humana ir a lugares que hasta ahora le estaban vedados, a lugares que según algunos quizá deberían estarles vedados. Es evidente que las personas ofendidas no creen que la inteligencia artificial ejerza la función liberadora de una escafandra autónoma. La inteligencia artificial amenaza de modo profundo y no muy sutil el concepto que tienen ellos de sí mismos: Nuestra identidad como persona deriva de nuestra inteligencia. La idea de que haya otras cosas —y peor aún de que haya cosas creadas por nosotros— que también puedan ser inteligentes requiere una revisión considerable de nuestra imagen.

Los científicos están creando máquinas cuyo objetivo es amplificar la inteligencia, una especie de escafandra autónoma.

Los intelectuales están experimentando de modo muy real e inmediato lo que muchos otros trabajadores han sufrido con anterioridad: la sustitución de sus habilidades especiales por máquinas. El profesor Edward Fredkin, del MIT, ofreció en cierta ocasión una perspectiva sobre el tema: «Los seres humanos no están mal. Estoy contento de ser uno de ellos. En general me gustan, pero sólo son humanos. Tampoco me quejaré. Las personas no son los mejores excavadores de zanjas del mundo, lo son las máquinas. Y las personas no pueden levantar los pesos que levanta una grúa. No pueden ni volar, si no se meten en un avión. Y no pueden transportar lo que transporta un camión. A mí esto no me ofende en absoluto. Hubo personas cuya vida giraba alrededor de algo exclusivamente físico: John Henry y el martillo de vapor. Ahora nos enfrentamos con el martillo de vapor intelectual. Al intelectual no le gusta que esta máquina lo haga mejor que él, y su reacción es idéntica a la del trabajador superado físicamente por la máquina.»[1]

Hay otras personas como Fredkin, cuyas identidades no se sienten amenazadas lo más mínimo por las posibilidades de una máquina inteligente. La trascendencia de un hecho así los emociona. No los preocupa mucho que el fenómeno se esté produciendo paso a paso y no de la noche a la mañana; esta lentitud sólo afecta a quienes no sólo desean que el fenómeno se produzca, sino que desearían acelerarlo porque hay muchas cosas por saber y por hacer y una máquina inteligente los ayudaría a saberlas y a hacerlas más rápidamente. También hay entre ellos personas en cuya opinión el término *inteligencia* está recargado de palabrería seudocientífica y que precisa algo de rigor empírico. En su opinión, asignar la palabra «inteligencia» al comportamiento de un ordenador no supone una apostasía excesiva. Y quizás en esto resida una de las explicaciones de la tranquilidad de McCorduck ante el hecho de la inteligencia artificial. La comprensión del tema no le llegó de repente. Empezó a darse cuenta mientras comparaba los argumentos en contra de las máquinas pensantes con las razones presentadas en el siglo XIX para demostrar que las mujeres no podrían estar nunca al mismo nivel intelectual que los hombres. McCorduck encontró paralelismos divertidos entre ambos argumentos. Al principio parecían únicamente un material jocoso y fácil de citar sobre la imposibilidad de que las mujeres pudiesen pensar realmente —razones de emoción, diferencias insuperables entre hombres y mujeres, la falta de precedentes, y además, sí, consideraciones éticas—, pero paulatinamente empezó a comprender que todo aquello ocultaba una verdad mayor. La *inteligencia* era un término político, definido por quienes mandan en cada momento. Esto explicaba su asombrosa elasticidad. Y entonces la cuestión de si una máquina puede pensar se convirtió de repente para McCorduck en una no cuestión, en un no tema absolutamente inconsecuente. Sin embargo, se plantea una cuestión más básica, a saber: ¿Importa realmente el ordenador? La respuesta es afirmativa, tanto en un sentido personal como global. El ordenador tiene importancia para el conjunto de nosotros, pero también tiene importancia para cada uno de nosotros. K64

LAS TS 2068 Y SPECTRUM POR DENTRO

Como ya lo habíamos mencionado en nuestro artículo anterior, ambas computadoras poseen el mismo microprocesador Z-80 de Zilog.
Este es un microprocesador de tercera generación: los microprocesadores de Intel 8008 y 8080 fueron sus antecesores inmediatos. La arquitectura interna del Z-80 es similar a la de estos micros. El set de instrucciones del Z-80 posee 158 tipos distintos contra las 58 instrucciones del 8008 y las 78 instrucciones del 8080.

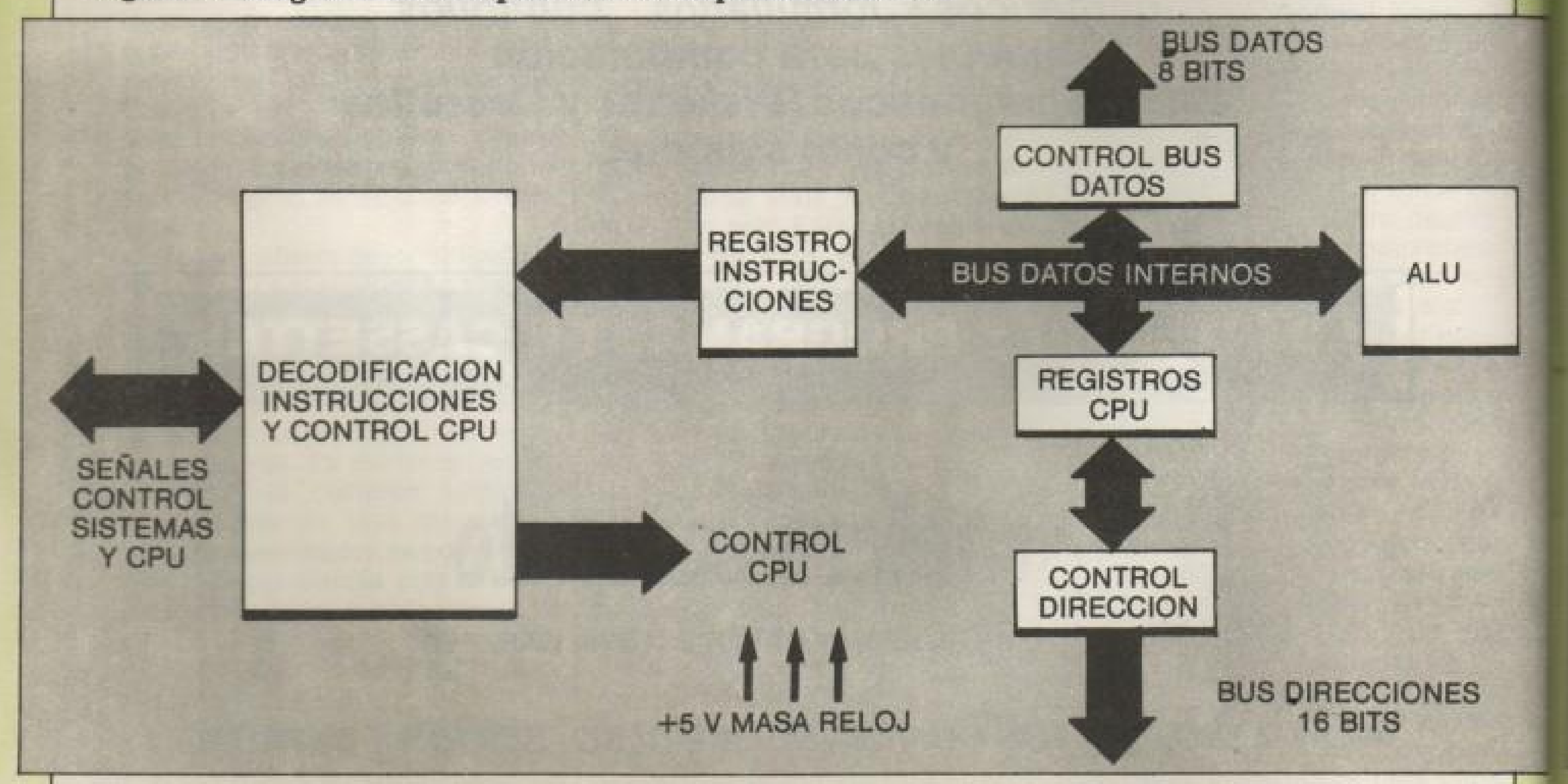
Arquitectura del Z-80

Según podemos observar en la figura 1, donde se representa el diagrama en bloques del Z-80, éste se encuentra organizado en distintos buses, siendo un bus un conjunto de conexiones eléctricas que cumplen una función similar. Por ejemplo, todos los datos que entran y salen del micro pasan por el bus de datos que, en este caso, por ser de ocho conexiones eléctricas, diremos que es un micro de ocho bits. El bus de direcciones es de 16 bits, por lo que el microprocesador podrá direccionar 2 Exp 16=65536 direcciones distintas de memoria o sea 64 Kbytes (1 K=1024 bytes). Internamente el Z-80 posee celdas de memoria llamadas registros de CPU (unidad central de proceso). Estos registros se pueden conectar con la Unidad Aritmética y Lógica (ALU) que realiza operaciones como ser suma, resta y operaciones lógicas. También existe un decodificador de instrucciones y controlador de la CPU el que a través de microinstrucciones programadas le irá diciendo a la CPU cómo debe ejecutar el programa en código de máquina e irá ordenando los distintos momentos en que se deben realizar las operaciones. Además el microprocesador puede recibir del exterior señales de control a través del llamado bus de control, al cual nos referiremos más adelante.

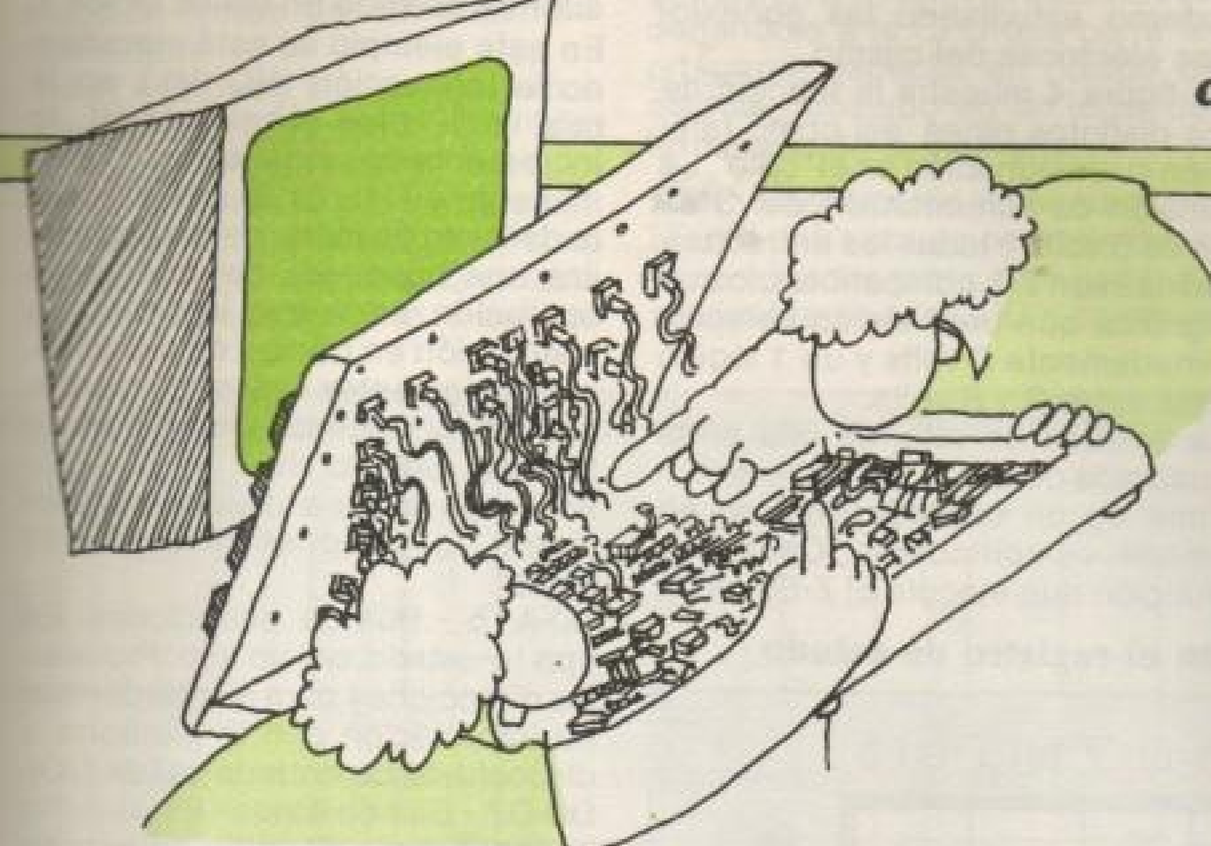
Registros del Z-80

Como se puede observar en la Figura 2, el Z-80 posee 18 registros de 8 bits y 4 registros de 16 bits. El principal registro utilizado para el manejo de datos es el registro A. Este nombre proviene del término registro de acumulador, y que en él se "acumulan" los resultados de operaciones aritméticas. Si, por ejemplo, dos operandos deben ser sumados, uno de ellos se coloca en el registro A, luego, los operandos son sumados por la ALU y el resultado de la suma se vuelve a almacenar en el acumulador. Ya que este registro es de ocho bits, sólo se podrán guardar números decimales de 0 a 255 (2 Exp 8=256).
Además del registro A, el Z-80 posee otros registros de uso general designados como B, C, D, E, H y L; estos registros también son de ocho bits y se utilizan para el almacenamiento temporario de valores. Por ejemplo, un número que se encuentra en el acumulador puede ser almacenado en el registro E a través de la instrucción LD E,A (cargar E con A). De esta forma se tiene una ventaja con respecto a la

Figura 1 Diagrama de bloques del Microprocesador Z-80



Ambas computadoras poseen el mismo microprocesador Z-80 de Zilog, al que le dedicaremos esta nota.



utilización de memoria externa: la CPU opera mucho más rápidamente con sus propios registros internos.
Como se puede observar en la figura, los registros se pueden agrupar de a pares: B con C, D con E y H con L, de esta forma algunas instrucciones del Z-80 permiten tratar a estos registros como de 16 bits. Por ejemplo, se podrán realizar operaciones aritméticas con números de 16 bits de largo.
El registro F, que se puede aparear con el acumulador A, es llamado Flag (bandera). En este registro queda almacenada información acerca de los resultados de ciertas operaciones. De los ocho bits del registro se utilizan seis. Cuatro de ellos se pueden interrogar y utilizarlos como condición para la ejecución de ciertas condiciones. Estos son la bandera de acarreo (C), la bandera de cero (Z), la bandera de signo (S) y la bandera de paridad/rebasamiento (P/V). La ubicación de estos bits dentro del registro F puede observarse en la Figura 3.
Además de estos ocho registros generales existe un juego duplicado de registros que se identifican con ('). De todas formas sólo un juego está activo al mismo tiempo, ya sea AF, BC, DE, HL o A'F', B'C', D'E', H'L' pero el programador tiene la posibilidad de intercambiarlos con las instrucciones de intercambio EX y EXX.
Por último el Z-80 posee seis registros de propósitos especiales.
Contador de programa (PC): Este registro contiene la dirección (16 bits) de la instrucción que en ese momento se está buscando en la memoria. Este contador es automáticamente incrementado después de que cada byte es leído. si pués de cada byte es leído.
Si se llegara a encontrar un salto o llamado a subrutina entonces la nueva dirección se sitúa en el PC.
Puntero de pila (SP): Es un registro de 16 bits que apunta a un sector de la memoria RAM que se encuentra reservada para el STACK (pila). El stack está organizado de forma tal, que los datos pueden ser almacenados y recuperados de la siguiente manera: el último que entra es el primero que sale (LIFO last-in, first-out). Esta área de memoria puede ser utilizada para guardar distintos valores cuando no se quieran usar los registros del microprocesador, pero además cumple una importante función, ya que en ella se almacenan las direcciones de retorno cuando se ejecutan

Figura 2 Configuración de registros de la CPU Z-80

REGISTROS DE USO ESPECIAL

VECTOR INTERRUPCION	REGENERACION DE MEMORIA R
REGISTRO INDICE IX	
REGISTRO INDICE IY	
PUNTERO DE PILA SP	
CONTADOR PROGRAMA PC	

REGISTROS DE USO GENERAL

CONJUNTO REGISTROS PRINCIPALES

ACUMULADOR	INDICADORES DE ESTADO
A	F
B	C
D	E
H	L

CONJUNTO REGISTROS ALTERNATIVOS

ACUMULADOR	INDICADORES DE ESTADO
A'	F'
B'	C'
D'	E'
H'	L'

instrucciones de llamado a subrutina o se retorna de una interrupción.
Registros índice (IX, IY): Son registros de 16 bits que permiten un tipo de direccionamiento de memoria llamado "indexado". Su utilización es muy común en el manejo de tablas.
Vector de interrupciones (I) y Registro de Refresco (R): El registro I se utiliza cuando el modo de interrupciones vectorizadas es elegido: suministra el byte alto de la dirección en la cual se realizará un llamado indirecto a una rutina de atención de interrupciones. El registro R se utiliza para el refresco de memorias RAM dinámicas.

Señales de Entrada-Salida

Pasemos ahora a considerar como el Z-80 se comunica con el mundo externo estudiando las conexiones eléctricas del mismo.
La figura 4 muestra la función de los distintos pines, así como también su designación en el "chip". La tensión de alimentación del Z-80 es de 5 volts y todas las entradas y salidas son TTL compatibles, lo que significa que un 0 lógico es aproximadamente 0 volts y un 1 lógico está entre 3 y 5 volts.
La entrada de reloj es una onda cuadrada de 3,528 MHz que se obtiene de un oscilador interno en ambas computadoras. Cada instrucción que ejecuta el Z-80 se divide en los llamados ciclos "T".
En la Figura 5 se puede observar como un ciclo de instrucción se encuentra dividido en varios ciclos T. En este ejemplo se está ejecutando la instrucción INC (HL), en la cual la CPU lee el valor de HL, lo incrementa en uno y lo vuelve a almacenar en HL. En la primera parte del ciclo de máquina la CPU realiza una búsqueda del código de operación, en la que interpreta la instrucción en código de máquina, en los siguientes ciclos realizará la lectura, incremento y nueva escritura del registro HL.
Pasemos ahora a describir brevemente las funciones de los demás pines:
A0-A15 - Bus de direcciones. Es tipo tri-estado, activo alto. Proveen las direcciones para el intercambio de información con la memoria o dispositivos de entrada-salida (I/O).
D0-D7 - Bus de datos - Estas ocho líneas bidireccionales tri-estado son utilizadas para la transferencia de datos con la CPU. La transferencia tiene lugar entre la memoria o los periféricos y el Z-80.
M1 - Ciclo de máquina 1 - Salida activa-baja. Esta señal significa que la CPU está obteniendo el código de operación para la próxima instrucción a ser ejecutada. M1 también aparece con IORQ para indicar un ciclo de reconocimiento de interrupciones.
MREQ - Memory request - Salida tri-estado, activa-baja. Esta señal le indica a la memoria que el bus de direcciones contiene una dirección válida para realizar una operación de lectura o escritura.
IORQ - Input/output request - Salida tri-estado, activa-baja. Indica que la mitad baja del bus de direcciones contiene una dirección válida de entrada-salida. Esta señal se utiliza para distinguir entre una operación de entrada-salida o una operación que involucra a la memoria. IORQ y MREQ nunca estarán activas al mismo tiempo.
RD - Read - Salida tri-estado, activa-baja. Indica que la CPU realizará una lectura de memoria o desde los dispositivos de entrada-salida.
WR - Write - Idem anterior, pero la CPU realizará una escritura.
RFSH - Refresco - Salida, activa-baja. Indica que los siete bits menos significativos del bus de direcciones contienen una dirección de refresco para memorias dinámicas.
HALT - Salida, activa-baja. Indica

Figura 3 Posiciones de los bits en el registro de estado

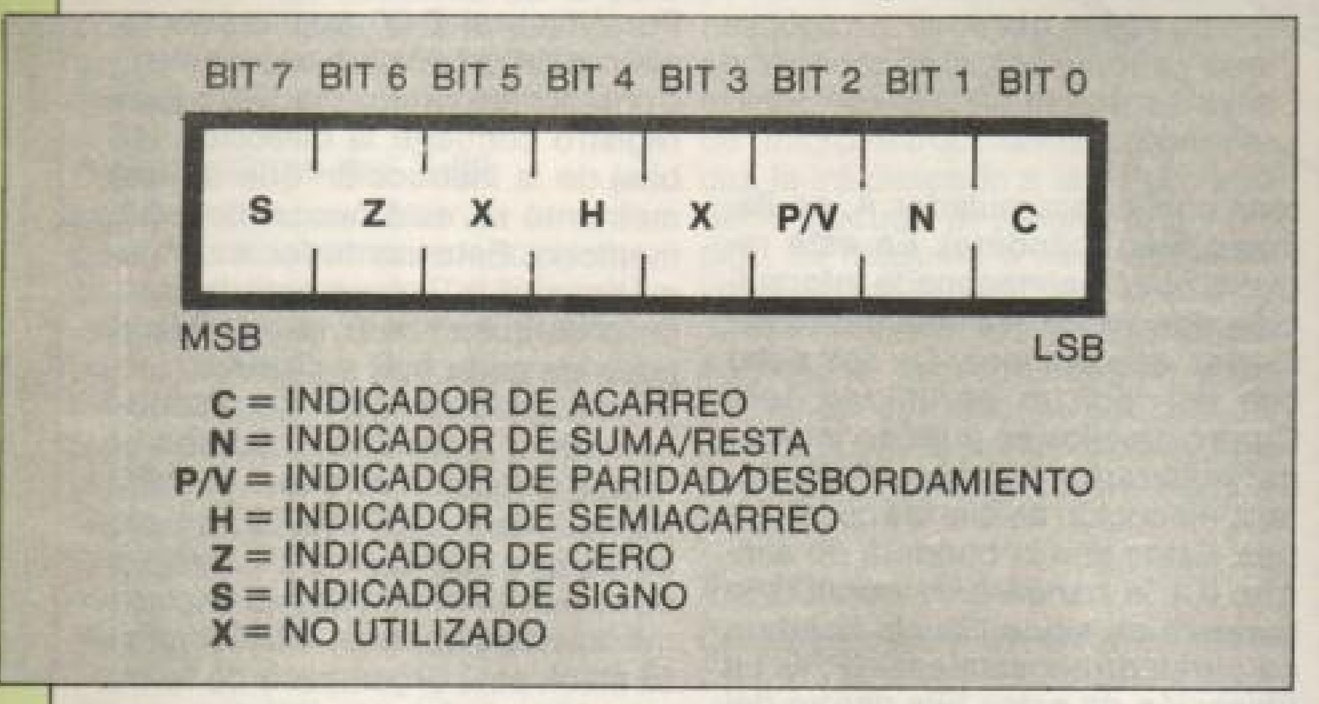
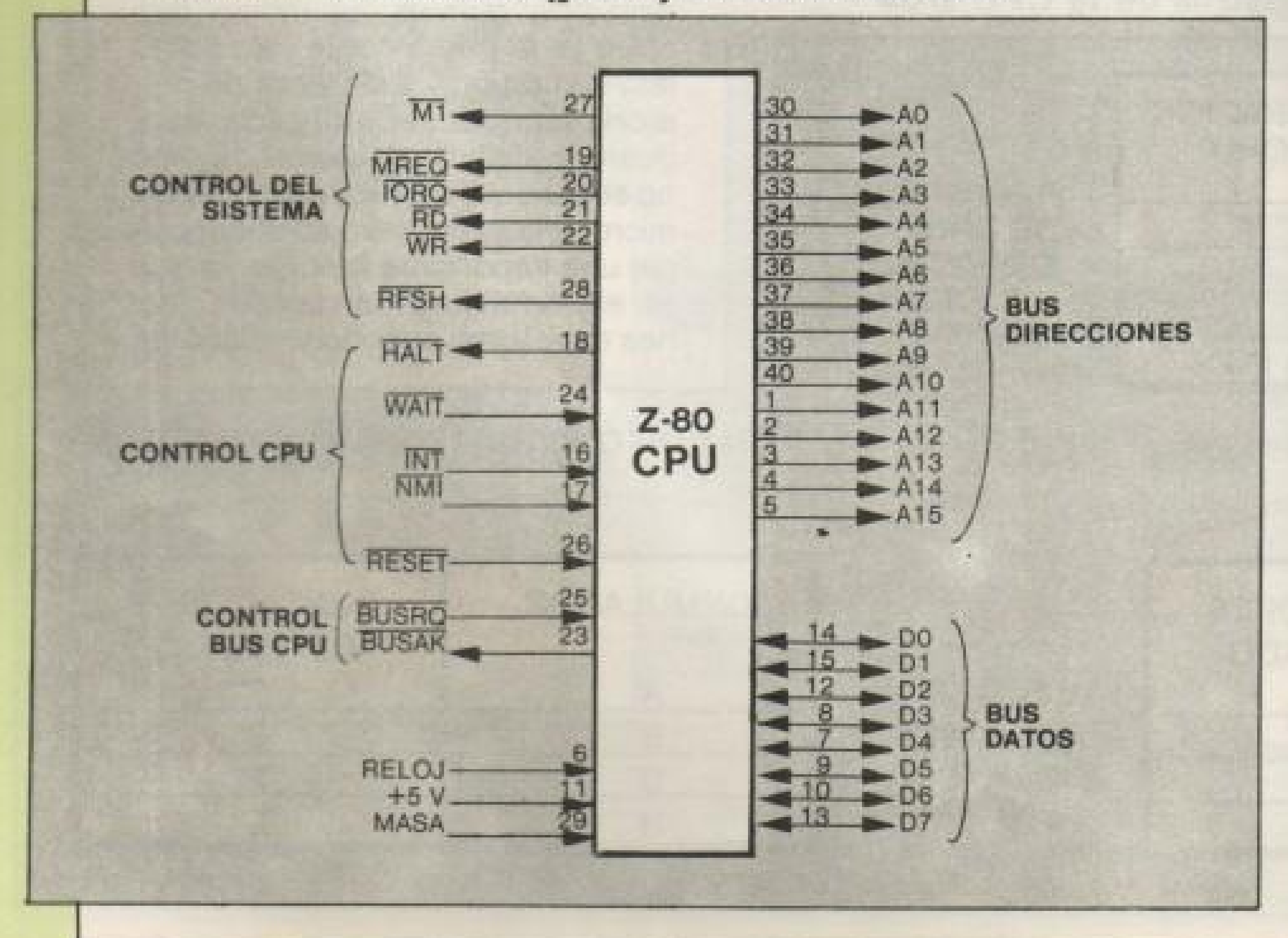


Figura 4 Terminales (pines) en el Micro Z-80



que la CPU ha ejecutado una detención por software provocada por una instrucción denominada HALT. La CPU esperará una interrupción antes de continuar su operación.

WAIT - Entrada, activa-baja. Indica a la CPU que la memoria o los dispositivos de entrada-salida direccionados no están aún listos para una transferencia de datos. La CPU no realiza ninguna operación hasta que el dispositivo lento le indique que está listo.

INT - Pedido de interrupción - Entrada, activa-baja. Esta señal se genera en dispositivos externos ordenándole a la CPU que corra un programa especial en código de máquina ubicado en un lugar determinado de la memoria.

NMI - Interrupción no enmascarable. Es una entrada operada por el flanco descendente de la señal. La misma fuerza a la CPU a correr un programa en código de máquina que comienza en la dirección decimal 102. A diferencia del otro tipo de interrupción, ésta no puede ser ignorada por la CPU. No se utiliza en estas dos computadoras.

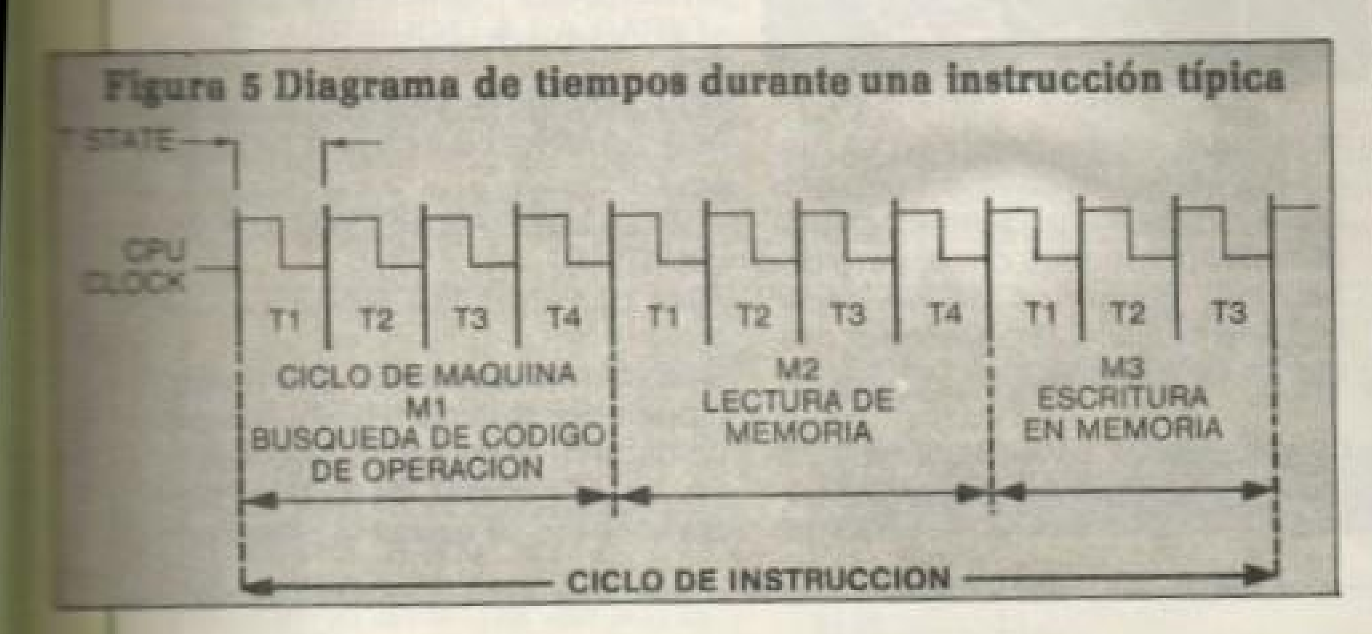
RESET - Entrada, activa-baja. esta señal inicializa la CPU borrando los registros PC, I y R. La CPU comenzará a ejecutar el programa a partir de la dirección 0 luego de un RESET.

BUSREQ - Solicitud de bus. Entrada, activa-baja. Es utilizada por un dispositivo externo a fin de solicitar el control de los buses. La CPU pasará sus buses al estado "tri-estado" cuando se termine el ciclo de máquina que se estaba ejecutando.

BUSACK - Salida, activa-baja. Indica al dispositivo solicitante, que puede tomar ahora el control de los buses.

De esta forma hemos dado un rápido vistazo al Z-80. Para interiorizarse más sobre el funcionamiento de este microprocesador, existe bibliografía disponible en librerías técnicas. **L. MATARRESE G.E.**

Figura 5 Diagrama de tiempos durante una instrucción típica

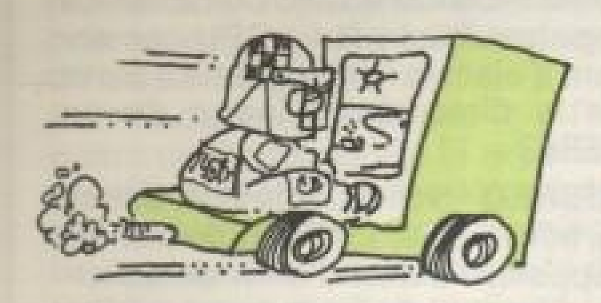


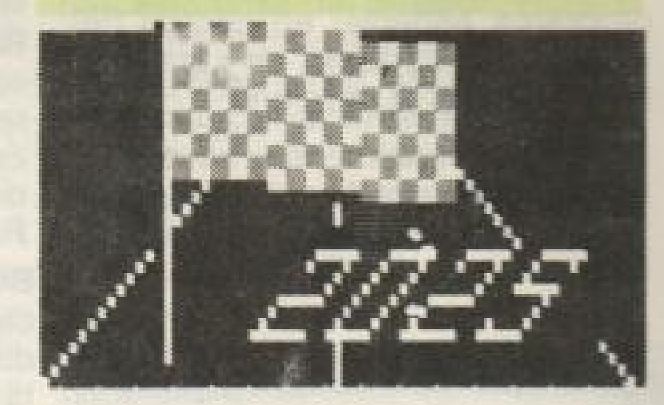


RALLY 2025



COMP.: TS 1000/1500; TK 83/85
CONF.: 16 K
CLAS.: ENT
AUTOR: Ricardo Alfredo Lucero
Córdoba





Al terminar de ingresar el programa en la computadora por primera vez, hay que hacerlo correr mediante RUN 4000 (RUN sólo no funcionará). La computadora se pondrá en modo FAST durante unos 120 segundos, regresando luego a SLOW para comenzar el juego.
La rutina 4000 se encarga de formar y guardar en memoria las pantallas que intervendrán en el juego. Por esto cuando vayamos a grabar el programa podemos borrarla para ocupar menos cinta; pero deberemos cuidarnos de no hacer CLEAR ni RUN. Grabémoslo mediante GOTO 4900.

En este juego nos disponemos a correr una etapa del "Rally del Futuro".

El piloto es el computador y nosotros los navegantes.
Por consiguiente deberemos darle las órdenes mediante teclado; y disponemos de los siguientes comandos:

- ACELERAR (produce un aumento de velocidad de 10 ó 15 kilómetros): Tecla "A".
- DISMINUIR (disminuye la velocidad en igual proporción): Tecla "D".
- FRENAR (reduce la velocidad en 60 kilómetros): Tecla "F".
- MARCHAS: Teclas "1" a "5".
- GIRAR A LA IZQUIERDA: "Tecla "7".
- GIRAR A LA DERECHA: Tecla "0".
- DESVIARSE A LA IZQUIERDA: Tecla "8".
- DESVIARSE A LA DERECHA: Tecla "9".

Para dar arranque pulsamos una tecla.
Deberemos controlar las revoluciones por minuto (RPM), las que no deberán bajar de 3500 (excepto en 1ª), ni superar las 6500. En el primer caso se apaga el motor con

```
   1 RETURN
  10 LET V=3
  20 LET TR=0
  30 LET K=0
  40 LET T=400
  50 LET A=0
  60 LET R=0
  70 LET M=1
  80 LET M1=1
  90 LET S=0
 100 LET B=1
 110 LET C=1
 120 PRINT AT 0,0;P$(1)
 130 PRINT AT 2,16;"████████████
█"
 140 IF INKEY$="" THEN GOTO 140
 150 PRINT AT 2,16;"████████████
█";AT 2,16;"███████████"
 160 IF INKEY$<>"" THEN GOTO 150
 190 PRINT AT 2,16;"████████████
█"
 200 LET M=3
 210 GOTO 230
 220 LET H=INT (RND*8)+3
 230 GOSUB 100*H
 240 LET C=1
 250 LET TR=TR+INT (T/50)
 260 LET K=K+6
 270 IF S=1 THEN GOTO 1940
 280 GOTO 220+1640*(A<35 AND M>1
)+1680*(A>65)+1880*(K>0)
 300 GOSUB 1100+30*(C>1)
 310 GOSUB 1620*((A$="7" OR A$="
0") AND A>0)+1550*(A$="9")+1300*
(A$="A")+1350*(A$="D")+1400*(A$=
"F")+1440*(CODE A$>=29 AND CODE
A$<=33)
 320 LET C=C+1
 330 IF S=1 OR C>6 THEN RETURN
 340 GOTO 300
 400 LET B=2
 410 GOSUB 1100+30*(C>1)
 420 GOSUB 1660*(A$<>"7" AND A$<
>"0" AND A$<>"" AND C=3)+1620*(A
$="0" OR (A$="" AND C=3))+1550*(
A$="9" AND C<3)+1480*(A$="7")+13
00*(A$="A" AND C<3)+1350*(A$="D"
 AND C<3)+1400*(A$="F" AND C<3)+
1440*(CODE A$>=29 AND CODE A$<=3
3 AND C<3)
 430 LET C=C+1
 440 IF S=1 OR C>3 THEN RETURN
 450 GOTO 410
 500 LET B=3
 510 GOSUB 1100+30*(C>1)
 520 GOSUB 1660*(A$<>"7" AND A$<
>"0" AND A$<>"" AND C=3)+1620*(A
$="7" OR (A$="" AND C=3))+1550*(
A$="9" AND C<3)+1480*(A$="0")+13
00*(A$="A" AND C<3)+1350*(A$="D"
 AND C<3)+1400*(A$="F" AND C<3)+
1440*(CODE A$>=29 AND CODE A$<=3
3 AND C<3)
 530 LET C=C+1
 540 IF S=1 OR C>3 THEN RETURN
 550 GOTO 510
 600 LET B=4
 610 GOSUB 1100+30*(C>1)
 620 GOSUB 1550*(A$="9")+1480*(A
$="7" OR A$="0")+1300*(A$="A")+1
350*(A$="D")+1400*(A$="F")+1440*
(CODE A$>=29 AND CODE A$<=33)
 630 LET C=C+1
 640 IF S=1 OR C>3 THEN RETURN
 650 GOTO 610
 700 LET B=4
 710 GOSUB 1100+30*(C>1)
 720 GOSUB 1660*(A$<>"7" AND A$<
>"0" AND A$<>"" AND C=3)+1620*(A
$="" AND C=3)+1550*(A$="9" AND C
<3)+1480*(A$="7" OR A$="0")+1300
*(A$="A" AND C<3)+1350*(A$="D" A
ND C<3)+1400*(A$="F" AND C<3)+14
40*(CODE A$>=29 AND CODE A$<=33
AND C<3)
 730 LET C=C+1
 740 IF S=1 OR C>3 THEN RETURN
 750 GOTO 710
 800 GOSUB 1100+30*(C>1)
 810 GOSUB 1620*(A$="7" OR A$="0
")+1550*(A$="9")+1300*(A$="A")+1
350*(A$="D")+1400*(A$="F")+1440*
(CODE A$>=29 AND CODE A$<=33)
 820 LET C=C+1
 830 GOTO 840+40*(S=1)-40*(C<4 A
ND S<>1)
 840 LET N1=5*(A>90 AND A<130)+2
*(A>=130)
 850 LET N=INT (RND*N1)
 860 IF N=1 THEN GOSUB 1800
 870 LET I$="LA VELOCIDAD NO ERA
 LA APROPIA- DA PARA PASAR UN VA
DO"
 880 RETURN
 900 GOSUB 1100+30*(C>1)
 910 GOSUB 1620*(A$="7" OR A$="0
")+1690*(A$="8" AND P=1)+1520*(A
$="8" AND P<>1)+1550*(A$="9")+13
00*(A$="A")+1350*(A$="D")+1400*(
A$="F")+1440*(CODE A$>=29 AND CO
DE A$<=33)
 920 LET C=C+1
 930 GOTO 940+20*(S=1 OR A$="8"
OR A$="9")-40*(C<4 AND S<>1)
 940 GOSUB 1800
 950 LET I$="LA PIEDRA ROMPIO EL
 EJE DEL AU- TOMOVIL"
 960 RETURN
1000 GOSUB 1100+30*(C>1)
1010 GOSUB 1620*(A$="7" OR A$="0
")+1520*(A$="8")+1550*(A$="9")+1
300*(A$="A")+1350*(A$="D")+1400*
```



```
(A$="F")+1440*(CODE A$>=29 AND C
ODE A$<=33)
1020 LET C=C+1
1030 GOTO 1040+20*(S=1 OR A$="8"
 OR A$="9")-40*(C<4 AND S<>1)
1040 GOSUB 1800
1050 LET I$="NO INTENTE EMULAR A
 LOS HEROES  DE LA TV. RESPETE L
OS CARTELES  DE LA RUTA"
1060 RETURN
1100 PRINT AT 0,0;P$(B);AT 0,4;A
;"█";AT 0,9;M;AT 0,14;R;"00";AT
0,21;K;AT 0,26;TR;AT 0,31;V
1110 IF M>6 THEN GOSUB 1060+M*20
1120 LET B=1
1130 PRINT AT C,0;"█"
1140 PAUSE T
1150 LET A$=INKEY$
1160 PRINT AT C,0;"█"
1170 RETURN
1200 PRINT AT 6,15;"██";AT 7,14;
"████";AT 8,13;"██████"
1210 RETURN
1220 PRINT AT 8,0;"██████████████
██████████████████████████████████
████████████"
1230 RETURN
1240 PRINT AT 18,19;"███";AT 19,
19;"████"
1245 LET P=INT (RND*2)
1250 IF P=1 THEN PRINT AT 17,9;"
███";AT 18,8;"████"
1255 RETURN
1260 PRINT AT 14,16;"  MANO    ";
AT 15,16;" CORTADA  ";AT 16,17;"█
██████";AT 17,17;"████████"
1270 RETURN
1300 LET A1=INT (RND*2)
1310 LET A=A+10*(A1=0)+15*(A1=1)
1320 LET T=T-20*(A1=0)-30*(A1=1)
1325 LET R=INT (A/M1)
1330 PRINT AT 0,4;A;"█";AT 0,14;
R;"00"
1340 RETURN
1350 LET A1=INT (RND*2)
1360 LET A=A-10*(A1=0 AND A>9)-1
5*(A1=1 AND A>14)
1370 LET T=T+20*(A1=0 AND A>9)+3
0*(A1=1 AND A>14)
1380 LET R=INT (A/M1)
1385 PRINT AT 0,4;A;"█";AT 0,14;
R;"00"
1390 RETURN
1400 LET A=A-60*(A>59)
1410 LET T=T+120*(A>59)
1420 LET R=INT (A/M1)
1425 PRINT AT 0,4;A;"█";AT 0,14;
R;"00"
1430 RETURN
1440 LET M=VAL A$
1450 LET M1=M/2+.5
1455 LET R=INT (A/M1)
```

la consiguiente pérdida de tiempo. En el segundo se funde el motor, con lo que se pierde un auto.

Al presentarse la situación tendremos la posibilidad de ejecutar hasta tres maniobras, y seis en recta. Pero si realiza la maniobra principal de la situación (por ej. doblar si hay una curva) en la primera oportunidad pasará de inmediato a la situación siguiente.

La computadora indica que está esperando la orden mediante una señal arriba a la izquierda. Al encenderse dicha señal digitemos la orden correspondiente. Si la señal se apaga antes de que hayamos pulsado algo habremos perdido esa oportunidad y vendrá la siguiente (si es que no era la última). El tiempo de espera de la computadora varía de acuerdo a la velocidad.

Las situaciones que se pueden presentar son: recta, curva, cruce o bifurcación (en ambas es válida cualquier dirección), vado (representado por una banda gris que cruza la ruta), piedras (manchas grises sobre la misma), o mano cortada.

Si perdemos los tres vehículos no podremos completar la etapa. En caso contrario encontraremos entre los 150 y 250 kilómetros la llegada. En ambos casos podremos observar los resultados finales, donde se indicarán los kilómetros recorridos, el tiempo empleado, el promedio y el puesto clasificatorio (esto último sólo si completamos la etapa).

A continuación nos preguntarán si deseamos correr otra etapa. Si contestamos que sí (S) nos pedirán nuevamente el nombre (pudiendo pulsar ENTER); y si contestamos no (N) aparecerá la pantalla final y se detendrá en un estado en el que sólo responderá a BREAK.

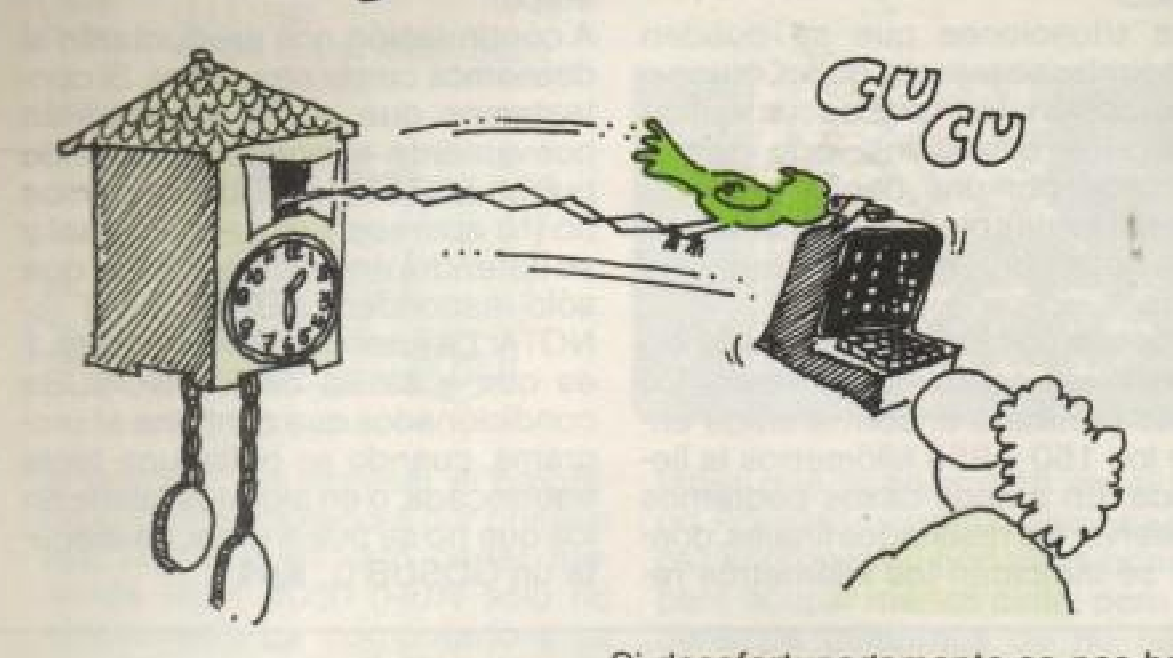
NOTA: La justificación de la línea 1 es que a causa de los GOSUBs condicionados que contiene el programa, cuando se pulsa una tecla equivocada, o en algunos casos en los que no se pulsa nada, se ejecuta un GOSUB 0. **K64**

```
1460 PRINT AT 0,9;M;AT 0,14;R;"0
0"
1470 RETURN
1480 LET N1=5*(A>110 AND A<140)+
2*(A)=140)
1490 LET N=INT (RND*N1)
1500 IF N=1 THEN GOSUB 1800
1510 LET I$="LA CURVA FUE TOMADA
 A EXCESIVA  VELOCIDAD"
1520 LET C=3
1530 RETURN
1550 LET N1=5*(A>100 AND A<130)+
2*(A)=130)
1560 LET N=INT (RND*N1)
1570 IF N=1 THEN GOSUB 1800
1580 LET I$="PISO LA BANQUINA A
MUY ALTA VE- LOCIDAD"
1590 LET C=C+2*(H=9 OR H=10)
1600 LET TR=TR+1
1610 RETURN
1620 GOSUB 1800
1630 LET I$="TOMO LA DIRECCION E
QUIVOCADA.   NO INTENTE SEGUIR P
OR DONDE NO  HAY CAMINO"
1640 RETURN
1660 GOSUB 1800
1670 LET I$="???????????????"
1680 RETURN
1690 GOSUB 1800
1700 LET I$="LAS PIEDRAS ESTABAN
 TAMBIEN SO- BRE LA CONTRAMANO.
EN UNA PRO-  XIMA SITUACION SIMI
LAR OBSERVE  MEJOR ANTES DE TIRA
RSE A LA IZ- QUIERDA"
1710 RETURN
1800 LET V=V-1
1805 LET S=1
1808 FOR G=0 TO 20
1810 FAST
1820 LET I=INT (RND*21)
1830 LET J=INT (RND*32)
1840 PRINT AT I,J;"▀▀▀▀▄▄▀▀▀▀▀▄▄
▄▄▀▀▀"
1845 SLOW
1850 NEXT G
1855 RETURN
1860 LET I$="LAS REVOLUCIONES BA
JARON Y SE   APAGO EL MOTOR"
1880 LET TR=TR+2
1890 GOTO 1940
1900 LET I$="UD NO CONTROLO DEBI
DAMENTE LAS  REVOLUCIONES Y FUND
IO EL MOTOR"
1920 LET V=V-1
1930 LET TR=TR+2*(V>0)
1940 CLS
1945 PRINT AT 7,0;"INFORME DE LA
 COMPUTADORA","-----------------
---------"
1950 LET X=10
1960 LET Y=0
1970 FOR G=1 TO LEN I$
1980 PRINT AT X,Y;" ";AT X,Y;I$(
G);"█"
1990 LET Y=Y+1
2000 IF Y=32 THEN GOTO 2060
2010 NEXT G
2020 PRINT AT X,Y;" "
2030 FOR G=0 TO 20
2040 NEXT G
2050 GOTO 40+2160*(V=0)
2060 LET X=X+1
2070 LET Y=0
2080 GOTO 2010
2100 PRINT AT 0,0;P$(1);AT 1,8;"
████████████";AT 2,8;"
iG▀▀▀▀ ";AT 3,8;"▄▄▄LI.ClILr
";AT 4,8;"
2110 FOR F=5 TO 10
2115 PRINT AT F,9;"█";AT F,21;"█
"
2120 NEXT F
2130 PRINT AT 18,0;"████████████
████████████████████████████████
████████████████████████████████
█"
2140 FOR F=0 TO 50
2145 NEXT F
2150 LET C$="COMPLETA"
2160 GOTO 2300
2200 PRINT AT 0,0;P$(1)
2230 PRINT AT 1,0;"█████████████
████████████████████████████████
████████████████████████"
2240 FOR F=0 TO 50
2250 NEXT F
2260 LET C$="INCOMPLETA"
2300 CLS
2310 PRINT "RESULTADOS FINALES",
"------------------",,,"PILOTO:
   Z-80",,"NAVEGANTE:      ";N$,"C
ONDICION: ETAPA ";C$,"KILOMETROS
 RECORRIDOS: ";K,"TIEMPO EMPLEAD
O: ";TR
2320 LET P=INT ((K/TR)*60)
2330 LET PC=0
2340 FOR F=1 TO 10
2350 IF PC>0 THEN GOTO 2370
2360 LET PC=F*(P<=145-10*F AND P
>=135-10*F)
2370 NEXT F
2380 LET B$=STR$ PC
2390 IF B$="0" OR CODE C$=46 THE
N LET B$="NO CLASI-FICO"
2400 PRINT "PROMEDIO: ";P;" KM/H
","PUESTO CLASIFICATORIO: ";B$
2410 PRINT AT 20,0;"DESEA CORRER
 OTRA ETAPA PARA     MEJORAR SU T
IEMPO?(S/N)"
2420 GOTO 2420+80*(INKEY$="N")-2
410*(INKEY$="S")
2500 CLS
2510 PRINT AT 0,0;X$(1 TO 32),,X
$(1 TO 64),,X$(1 TO 96),,X$(1 TO
 128),,X$(33 TO 128),,X$(65 TO 1
28),,X$(97 TO 128)
2520 GOTO 2520
3000 PRINT AT 0,0;P$(1)
3010 FOR F=0 TO 5
3020 PRINT AT F,0;"█████████████
███████████████████"
3030 NEXT F
3040 PRINT TAB 7;"▒ ▒ ▒ ▄ ▄ "
3050 FOR F=0 TO 2
3060 PRINT TAB 7;"▒ ▒ ▀▀▀▀▒ ▒
▒";TAB 7;"▒ ▒ ▀▀▀▀ ▒ ▒ "
3070 NEXT F
3080 PRINT TAB 12;"▄▄▄▄▄▒ ▒ ▒"
3090 FOR F=6 TO 20
3100 PRINT AT F,6;"█"
3110 NEXT F
3120 PRINT AT 1,0;X$;AT 16,11;"█
███████████";AT 17,11;"█
███████████";AT 18,11;"█
███████████";AT 19,11;"
"
3130 FOR F=0 TO 80
3140 NEXT F
3150 CLS
3160 PRINT AT 21,0;"████████████
█████"
3170 INPUT M$
3180 IF M$<>"" THEN LET N$=M$
3190 LET Q=INT (RND*100+150)
```



```
3200 GOTO 10
4000 FAST
4010 DIM P$(4,704)
4020 LET X$="███████████████████
████████████████████████████████
"
4030 FOR F=0 TO 21
4040 PRINT "████████████████████
████████████"
4050 NEXT F
4060 FOR F=0 TO 31
4070 UNPLOT F,F
4080 UNPLOT 62-F,F
4090 NEXT F
4100 PRINT AT 6,0;"▄▄▄▄▄▄▄▄▄▄▄▄▄▄
▄▄█▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀"
4110 PRINT AT 0,0;"█████████████
███████████████████"
4120 FOR F=8 TO 9
4130 PRINT AT F,15;"█";AT F+4,15
;"█";AT F+10,15;"█";AT F+12,15;"
█"
4140 NEXT F
4150 PRINT AT 14,15;"█"
4160 LET N=1
4170 GOSUB 4530
4180 GOSUB 4600
4190 PRINT AT 6,15;"▄▄";AT 9,12;
"████";AT 7,0;"▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀
";AT 8,0;"▀▀▀▀█▀▀█▀▀▀▀█";AT 1
0,0;"▀▀▀▀▀▀▀▀▀█"
4200 GOSUB 4500
4210 GOSUB 4600
4220 PRINT AT 0,0;P$(1)
4230 PRINT AT 6,15;"▄▄";AT 9,15;
"████";AT 7,15;"▀▀▀▀▀▀▀▀▀▀▀▀
▄▄";AT 8,15;"█▀▀█▀▀█▀▀█▀▀";
AT 10,19;"█▀▀▀▀▀▀▀▀▀"
4240 GOSUB 4500
4250 GOSUB 4600
4260 PRINT AT 0,0;P$(1)
4270 PRINT AT 8,0;"▀▀▀▀▀▀▀▀▀▀▀▀▀
";AT 8,18;"▀▀▀▀▀▀▀▀▀▀▀▀";AT 9,
12;"██████";AT 10,0;"▄▄█▀▀█▀
█▀";AT 10,19;"█▄▄█▀▀▀▀█";AT
 11,0;"▀▀▀▀▀▀▀█";AT 11,20;"█
▀▀▀▀▀▀▀"
4280 GOSUB 4500
4290 GOSUB 4600
4300 CLS
4340 SLOW
4350 GOTO 3000
4500 FOR F=3 TO 5
4510 PRINT AT F,0;"█████████████
███████████████████"
4520 NEXT F
4530 LET H=INT (RND*3)+3
4540 FOR F=1 TO H
4550 LET B=INT (RND*27)+1
4560 IF B=14 THEN GOTO 4550
4570 PRINT AT 3,B;"▛ ▘";AT 4,B;"
▙ ▖";AT 5,B;"█▀"
4580 NEXT F
4590 RETURN
4600 LET H=PEEK 16396+256*PEEK 1
6397
4610 FOR F=1 TO 704
4620 LET L=PEEK H
4630 IF L=118 THEN GOTO 4690
4640 LET P$(N,F)=CHR$ L
4650 LET H=H+1
4660 NEXT F
4670 LET N=N+1
4680 RETURN
4690 LET H=H+1
4700 GOTO 4620
4900 SAVE "RALLY 202█"
5000 GOTO 3000
```

RELOJ DIGITAL

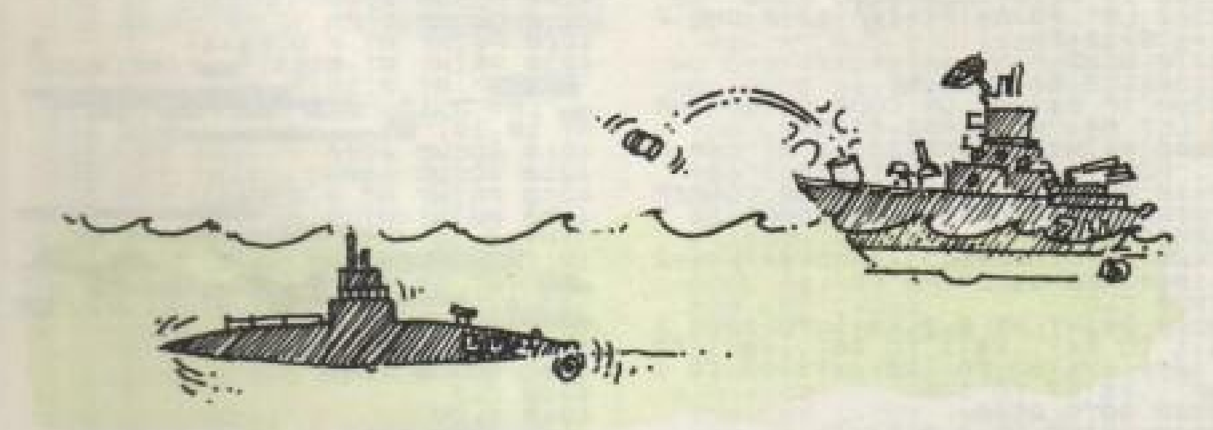


Si desafortunadamente se nos ha roto el reloj, con este programa bien cortito podremos disponer de un reloj digital. Tal vez no resulte muy práctico, pero funciona. Deberemos arrancar con RUN.

COMP.: CZ 1000/1500; TK 83/85
CONF.: 2 K
CLAS.: UTI

```
  1 REM RELOJ DIGITAL
  4 LET L=1
  5 PRINT "HORA ?"
 10 INPUT H
 15 PRINT "MINUTO ?"
 20 INPUT M
 25 CLS
 35 IF M=60 THEN GOTO 100
 39 PRINT AT 10,10;H;":";M
 50 LET M=M+L
 60 FOR F=L TO 2732
 70 NEXT F
 80 GOTO 30
100 LET M=0
105 LET H=H+L
110 IF H>12 THEN LET H=
120 CLS
130 GOTO 39
```

SUBMARINO II



Estamos conduciendo un submarino y deberemos atravesar el mar esquivando las peligrosísimas minas. Se arranca con RUN y se maneja con las flechas.



COMP.: CZ 1000/1500, TK 83/85
CONF.: 2 K
CLAS.: ENT

```
  1 REM SUBMARINO II
  5 FOR F=SGN PI TO INT PI
 10 CLS
 15 FAST
 20 PRINT "---------------------
-------------";AT VAL "8",NOT PI,
"███████████████████████████████
█"
 25 FOR M=NOT PI TO CODE "2"
 30 PRINT AT VAL "6"*RND+SGN PI
,RND*CODE "?"+CODE "▌";"*"
 35 NEXT M
 40 SLOW
 45 LET D=INT (VAL "6"*RND+SGN
PI)
 50 FOR S=NOT PI TO CODE ";"
 55 PRINT AT D,S;
 60 IF PEEK (PEEK 16398+PEEK 16
399*256)=CODE "*" THEN PRINT "▘▖
";TAB NOT PI;"DESTRUIDO ";F;AT D
,S;" ";
 65 PRINT "█"
 70 LET A=D
 75 LET D=D+(INKEY$="6" AND D<V
AL "7")-(INKEY$="7" AND D>SGN PI
)
 80 PRINT AT A,S;" "
 85 NEXT S
 90 NEXT F
 95 PRINT "BIEN, ATRAVESASTE EL
PELIGRO"
```

RENUMERADOR



COMP.: CZ 1000/1500, TK 83/85
CONF.: 2 K
CLAS.: UTI



Con este programa de pocos pasos, se puede poner un poco de orden en nuestros revueltos programas. Sobre todo en aquellos que por teclearlos en pasos de a uno, no nos queda lugar en el medio de dos pasos para poder meterse. Se arranca con GOTO 9000.

```
9000 REM RENUMERADOR
9001 LET IO=10
9010 FAST
9020 LET L=16511
9030 IF PEEK (L-2)>=35 THEN GOTO
 9100
9040 LET H=PEEK L+256*PEEK (L+1)
9050 POKE (L-2),INT (IO/256)
9060 POKE (L-1),IO-INT (IO/256)*
256
9070 LET IO=IO+10
9080 LET L=L+H+4
9090 GOTO 9030
9100 SLOW
9110 LIST
```

CONSTRUCTOR

COMP.: CZ 1000/1500, TK 83/85
CONF.: 2 K
CLAS.: ENT



Deberemos empujar los signos "=", apretando cualquier tecla, y apilarlos línea a línea. Si soltamos la tecla, el signo se detiene, y ubicando el cuadrado negro, sabremos dónde nos encontramos. Para cargarlo tardaremos menos de cinco minutos, pero horas para jugarlo. ¡Animo!

```
  1 GOTO 10
  2 SAVE "CONSTRUCTO█"
 10 FOR I=0 TO 21
 20 PRINT AT I,25;"█"
 30 FOR J=0 TO 23
 40 PRINT AT I,J;">=" AND INKEY
$<>""
 50 NEXT J
 60 PRINT AT I,25;" "
 70 NEXT I
 80 GOTO 0
```

DESENSAMBLANDO EL ASSEMBLER

Eduardo H. Mombello

PARTE II

Veremos ahora el último de los registros y sus dos instrucciones asociadas, antes de comenzar a programar.

El registro SP (Stack Pointer)

El SP o puntero de la pila, es un registro un tanto especial de 16 bits, veamos su funcionamiento.
La pila a la que hace referencia es un conjunto de bytes consecutivos, que los podemos pensar como cajitas una encima de otra, de ahí el concepto de pila de bytes.
Entonces la cajita que se encuentra en la cima de la pila será la posición de memoria de menor valor, y el SP contendrá el valor de dicha posición, aclaremos esto,

```
SP  _ [56h] _15000.
SP+1_ [A2h] _15001.     Posiciones
SP+2_ [05h] _15002.         de
SP+3_ [67h] _15003.      memoria
       "pila"
```

En el cuadro anterior vemos que la "pila" comienza en la posición de memoria 15000. y que el SP apunta a esta dirección, también se ve que dicha posición contiene el valor 56 hexadecimal.
Cabe preguntarnos para qué sirve ésto, pues bien, se lo utiliza para guardar datos que nos servirán más adelante justamente apilados como si fueran libros.

PUSH y POP

Por fin llegamos a ver nuestras primeras instrucciones, hablemos de PUSH.
Esta nos permite guardar datos contenidos en cualquiera de los pares BC, DE, HL, AF, IX e IY, en la cima de la pila a la que apunta el SP.
Un dato importante es que el contenido del registro que guardamos en la "pila" permanece en dicho registro sin ser afectado.
Veamos un ejemplo:
Supongamos que nuestro SP está en las condiciones del cuadro que vimos antes, y que el par DE contiene el número 16384, o sea por lo que vimos en el número anterior el registro D contendrá el valor 64. y el E tendrá un 0.
Entonces veamos cómo quedará nuestra "pila" una vez que ejecutemos la instrucción PUSH DE.

```
SP  —— [ 0. ]  14998.
SP+1— [ 64. ]  14999.   Posición
SP+2— [ 56h ]  15000.      de
SP+3— [ A2h ]  15001.    memoria
SP+4— [ 05h ]  15002.

        "pila"

    D [ 64. ]   Contenido
    E [ 0.  ]      del
                 par DE.
```

Ahora tenemos una copia del contenido del par DE encima de la pila, notar que nuestra pila comienza ahora en la posición de memoria 14998, con lo cual si examinamos el contenido del SP veremos que justamente ha tomado ese valor.
Otra cosa importante de observar es que guarda "primero" el contenido del registro de menor peso y luego el de mayor.
En nuestro caso, el contenido de E se encuentra en el lugar a donde apunta SP, y el de D está a donde apunta SP+1.
A partir de este momento cuando encerremos entre paréntesis a cualquiera de los registros dobles o a un número que represente una posición de memoria, estaremos refiriéndonos al comienzo de dicha posición de memoria o al contenido de la posición de memoria que contiene el registro.
Ejemplo: si el par BC contiene el número 65535, y en la posición de memoria 65535. se encuentra el número 12. se tendrá:
BC=65535. (BC)=12. y
(BC)=(65535.)=12.
Si, en cambio, ahora, queremos guardar en uno de los registros dobles lo que se encuentre en la cima de nuestra pila, que en definitiva es extraer el contenido de los dos bytes superiores de la pila, debemos ejecutar la instrucción POP rr, donde rr es cualquiera de BC, DE, HL, AF, IX e IY.
Como ejemplo podemos ver que, si tenemos la memoria de nuestro computador cargada con los valores del ejemplo anterior, al igual que el SP apuntando a la posición 14998., luego de hacer la extracción por medio de POP HL el SP apuntará a la posición 15000.
El par HL tendrá el valor 16384., entonces nuestra representación de las celdas de memoria será:

```
No
pertenecen
al         [ 0.  ] 14998.
stack.     [ 64. ] 14999.    Posición
SP         [ 56h ] 15000.       de
SP+1       [ A2h ] 15001.    memoria
SP+2       [ 05h ] 15002.
```

Figura 1

Posición de memoria	CM	Mnemónico	Notas
40001	245.	PUSH AF	Guarda los registros
40002	197.	PUSH BC	en la pila.
40003	213.	PUSH DE	.
40004	229.	PUSH HL	.
40005/6	62. 170.	LD A,170.	A 170.
40007/8/9	33. 110. 75.	LD HL,19310.	HL 19310.
40010	119.	LD (HL),A	19310. A
40011	225.	POP HL	Extrae los valores
40012	209.	POP DE	de los registros
40013	193.	POP BC	de la pila.
40014	241.	POP AF	
40015	201.	RET	Retorna al BASIC

En esta nota terminamos de explicar los registros y empezamos a ver algunas instrucciones.



Entonces podemos decir que POP hace el camino inverso al de PUSH.

Grupo de carga de 8 bits

Antes que nada dejaremos en claro algunas cosas:

I) Ninguna de las instrucciones que pertenecen a este grupo afecta al indicador de estado F (flag).

II) r y r' son cualquiera de los registros A, B, C, D, E, H o L.

III) n es un número cualquiera representable en 8 bits (0. a 255.).

IV) nn es un número representable en 16 bits (0. a 65535.).

V) (nn) indica el contenido de la posición de memoria representada por nn.

VI) El CM (código máquina) de cada instrucción se dará a medida que lo requieran los ejemplos, de todos modos en el manual de la TS 2068 pueden encontrarlos en el apéndice B.

Aclarados estos puntos comencemos con las instrucciones.

LD r,r': ésta copia el contenido de r' en el registro r.

LD r,n: r toma el valor n.

LD r,(HL): r toma el valor que está guardado en la posición a la que apunta HL.

LD (HL),r: copia en la posición de memoria a la que apunta HL el número que contiene r.

LD (HL),n: copia n en (HL).

LD A,(BC): copia en el acumulador (BC).

LD A,(DE): copia en el acumulador (DE).

LD A,(nn): copia en el acumulador (nn).

LD (BC),A: copia en (BC) el contenido de A.

LD (DE),A: copia en (DE) el contenido de A.

LD (nn),A: copia en (nn) el valor de A.

Estas son las que más nos interesan en este grupo, pues son las de uso más corriente.

Ahora sí, programemos algo y analícemos en detalle su funcionamiento.

En primer lugar debemos elegir un sitio seguro para alojar nuestro programa.

En el caso del Spectrum y de la 2068 existe una instrucción Basic (CLEAR) que seguida de un número correspondiente a una posición de memoria, le indica al sistema operativo que cualquier programa basic que se intente ingresar no podrá sobrepasar dicha posición de memoria.

Por ejemplo, si ingresamos CLEAR 40000 nos quedarán 25 kbytes después de la posición 40000, para guardar "cosas" que no sean un programa basic.

En nuestro caso lo usaremos para guardar el CM.

Es importante también, que leamos del manual de nuestra computadora cómo se encuentra repartido el lugar de trabajo del sistema operativo, por el momento busquemos el comienzo y fin de la memoria de pantalla.

En el caso del TS 2068 y del Spectrum ésta comienza en la posición 16384. y termina en la 22527.

En primer lugar, podemos ver que el programa de la figura 1 utiliza instrucciones que conocemos, veamos qué es lo que hacen.

Los cuatro primeros PUSH guardan los valores que tenían los registros antes de la ejecución de nuestra rutina en CM.

Luego, se carga el acumulador con el valor 170.(10101010) y se copia este número en la posición de memoria 19310.

Esta última, como podemos ver, se encuentra dentro del rango de memoria destinado a la pantalla.

Por último, los cuatro POP devuelven su antiguo valor a los registros, y RET finaliza nuestro programa entregando el control al sistema operativo del Spectrum ó 2068.

Al ejecutar esta rutina (con RANDOMIZE USR 4e4) sólo veremos cuatro puntitos en medio de la pantalla (posición 19310.) provenientes del número 170.(10101010).

Lo importante es que nos detengamos a entender cada punto del programa (figura 1), y ver su forma de llevarlo a la práctica (figura 2) por medio del basic.

Ahora veremos otra rutina (figura 3) que llenará la pantalla con "basura", pero que tiene por finalidad hacer la comparación del tiempo que tarda en Assembler un bucle (FOR) de 6911 pasos, con lo que tardaría en terminar el mismo bucle pero en basic.

Para ver este efecto compilemos el CM de la figura 3, de forma similar al de la figura 2, y ejecutémoslo (RANDOMIZE USR nn). Luego carguemos el basic de la figura 4 y ejecutémoslo (RUN).

Si prestamos atención a la ejecución del programa en basic, notaremos la forma en que se distribuyen los bytes de pantalla, es lento ¿no? K64

Figura 2

```
10 CLEAR 40000
20 FOR f=40001 TO 40015: READ a
30 POKE f,a: NEXT f
40 DATA 245,197,213,229,62,170,33,110,75,119,225,209,193,241,
   201
```

Figura 3

Posición	CM	Mnemónico
40001/2/3	33. 0. 0.	LD HL,0.
40004/5/6	17. 0. 64.	LD DE,16384.
40007/8/9	1. 255. 26.	LD BC,6911.
40010/11	237. 176.	LDIR
40012	201.	RET

Figura 4

```
10 FOR f=16384 TO 16384+6911: POKE f,PEEK (f-16384): NEXT f
```

EL SISTEMA TRIDIMENSIONAL

En el libro "Ideas y formas. Explorando el espacio con Logo", del ingeniero Horacio Reggini (Ed. Galápago, Dist. Emecé) el autor explica la creación y el manejo de formas tridimensionales.

El sistema tridimensional básico es el más rápido y simple. Contiene los procedimientos necesarios para mover la tortuga tridimensional y realizar figuras espaciales (ANDAR, ROLAR, CABECEAR, ROLAR, TRI).

SPECTRUM CZ-2000

```
TO ANDAR: D
MAKE "X (:X+:D*:C11)
MAKE "Y(:Y+:D*:C12)
MAKE "Z(:Z+D*:C13)
GRAFICAR :X :Y :Z
END

TO VIRAR A
MAKE "C 005 :A
MAKE "S SIN :A
MAKE "T11(:C11*:C+:C21*:
S)
MAKE "T12 (:C12*:C+:C22*:
S)
MAKE "T13(:C13*:C+:C23*:
S)
MAKE "T21(:C21*:C-:C11*:
S)
MAKE "T22(:C22*:C-:C12*:
S)
MAKE "T23(:C23*:C-:C13*:
S)
REEMPLAZAR
END

TO CABECEAR :A
MAKE "C COS :A
MAKE "S SIN :A
MAKE "T11(:C11*:C-:C31*:
S)
MAKE "T12(:C12*:C-:C32*:
S)
MAKE "T13(:C13*:C-:C33*:
S)
MAKE "T31(:C31*:C+:C11*:
S)
MAKE "T32(:C32*:C+:C12*:
S)
MAKE "T33(:C33*:C+:C13*:
S)
REEMPLAZAR
END

TO ROLAR :A
MAKE "C COS :A
MAKE "S SIN :A
MAKE "T21(:C21*:C+:C31*:
S)
MAKE "T22(:C22*:C+:C32*:
S)
MAKE "T23(:C23*:C+:C33*:
S)
MAKE "T31(:C31*:C-:C21*:
S)
MAKE "T32(:C32*:C-:C22*:
S)
MAKE "T33(:C33*:C-:C23*:
S)
REEMPLAZAR
END

TO REEMPLAZAR
MAKE "C11 :T11
MAKE "C12 :T12
MAKE "C13 :T13
MAKE "C21 :T21
MAKE "C22 :T22
MAKE "C23 :T23
MAKE "C31 :T31
MAKE "C32 :T32
MAKE "C33 :T33
END

TO GRAFICAR :U :V :W
SETPOS SE (:F*:U/(:F-:W
))(:F*:V/(:F-:W))
END

TO TRI
SETBG 6 CS CT HT WINDOW
MAKE "F 360
MAKE "T11 0
MAKE "T12 1
MAKE "T13 0
MAKE "T21 -1
MAKE "T22 0
MAKE "T23 0
MAKE "T31 0
MAKE "T32 0
MAKE "T33 1
MAKE "X 0
MAKE "Y 0
MAKE "Z 0
REEMPLAZAR
END
```

Esta versión del sistema no hace aparecer en la pantalla la tríada —conjunto de tres ejes que muestran la ubicación y orientación de la tortuga tridimensional—; tampoco incorpora las órdenes de movimientos relativos a los ejes coordenados fijos.

El sistema tridimensional completo, además de los procedimientos necesarios para mover la tortuga tridimensional y realizar figuras espaciales, tiene la capacidad de hacer aparecer en la pantalla la tríada. Esta versión del sistema incorpora a las órdenes del sistema básico aquellos procedimientos que permiten la manipulación de la tríada (TRIINV, TRIADA, SINTRIADA, VISIBLE, INVISIBLE). Las órdenes de movimiento, tanto del sistema básico como del sistema completo, están referidas a la geometría intrínseca de la tortuga tridimensional; no han sido incorporadas las órdenes de movimientos relativos a los ejes coordenados.

El sistema adicional incorpora órdenes espaciales referidas a la geometría extrínseca de la tortuga tridimensional; o sea, los procedimientos que definen las ubicaciones de la tortuga tridimensional en relación a los ejes coordenados (PX3D, PY3D, PZ3D, PXYZ3D, POS3D, CENTRO3D, PUNTO3D), que son similares a las usadas por la tortuga bidimensional en el plano.

Esta versión incluye también órdenes de rotación de la tortuga alrededor de los ejes fijos (GIRARX, GIRARY, GIRARZ).

SPECTRUM CZ-2000

```
TO RECTANGULO
REPEAT 2 (ANDAR 10 VIRAR 90 ANDA
R 120 VIRAR 90)
END

TO MOLINETE
REPEAT 18 (RECTANGULO ROLAR 20)
END
```

Por primera vez en Argentina

K64 transmite en LASER 102

Programas para Home Computer
Todos los jueves a las 6,40 Hs.

TRANSMISION ENERO 1986

2/1/1986: CHECQUERED FLAG

La emoción de la fórmula 1 en tu computadora. Se te ofrece la posibilidad de correr en algunos de los más famosos circuitos del mundo. El conducir un auto de estas características, es una tarea de habilidad en la que se requiere evitar riesgos, permanecer en pista, cuidar el vehículo y lograr nuevos records!

Comandos:
Acelerador: O
Freno: I
Rebaje: N
Girar izquierda rápido: A
Girar izquierda lento: S
Girar derecha rápido: F
Girar derecha lento: D
Pausa: H
Finalizar: H y T a la vez

Hay que cuidarse de los siguientes riesgos:
Agua y aceite en pista; Vidrios; Recalentamiento del motor; Nivel de nafta; etc.
Para los más inexpertos, se puede comenzar con el "Mc Faster Special", que tiene caja automática.

9/1/1986: MONOPOLIO

Este juego es parecido al tradicional "Estanciero". Permite el juego de 2 a 6 jugadores. Al comienzo cada uno recibe un informe del estado de caja y propiedades que posee.

Comandos:
LIST: Lista todas las propiedades que posee.
H: Le permite vender sus propiedades al banco o hipotecarias.
B: Compra propiedades de otro jugador al precio convenido.
S: Vende propiedades a otro jugador.
V: Para recuperar propiedades del banco hipotecadas.
Q: Para el juego mostrando el jugador. Se puede en ese momento, guardar el juego en cassette hasta donde se llegó.
T: Graba el programa en cassette.
R: En caso de haberse interrumpido el programa, arrancarlo con GOTO D y luego pulsar R.
M: Cuando un jugador hipoteca, recibe la mitad del valor de su propiedad. Para recobrarla con este comando, se deducirá de sus reservas de caja esa cantidad más un 10%.
DOBLES: Si sale dobles en los dados, puede jugar otra vez pero si sale doble una tercera vez irás a la cárcel permaneciendo allí 3 jugadas. (Se puede salir pagando 5000).
Demás instrucciones durante el juego.

16/9/1986: ANT ATTACK

La ciudad de "Antchester" ha sido abandonada por muchos años y permanece copada por gigantescas hormigas asesinas. Pero como siempre ocurre en toda aventura, en esta no podían faltar los héroes, que por primera vez en los juegos electrónicos, en este se puede elegir "héroe" o "heroína"!
El objeto del juego es que "el" o "ella" rescate al otro buscándolo entre las ruinas de la ciudad y salir cuanto antes salvándose de las mordidas de las hormigas asesinas.

Comandos:
O,P, ENTER, SPACE: Cambia ángulo de visión
SYMBOL SHIFT: Rota a reloj
M: Rota contra reloj
V: Avanza
C: Salta
S,D,F,G: Tira granadas de distintos alcances
1: Ultima oportunidad, vuelve a la entrada.

23/1/1986: SUPERGRAF

Este es un programa del tipo educativo, que permite realizar gráficos en baja, media y alta resolución de pantalla.
Es muy fácil de manejar, y se muestran varios ejemplos logrados con él.

Comandos:
T: Color de tinta
D: Ancho del trazo de 1 a 3
B: Borra
C: Acceso a líneas curvas
R: Acceso a líneas rectas
O: Borra pantalla
L: Inserta texto
G: Cambia cursos a modo G de 1 a 8
S: Salta de un punto a otro
P: Promoción de líneas curvas
O: OVER-1 (sobre impresión)
N: OVER-0 (normal)
F: Fin del dibujo y acceso a cassette
Para las curvas, hay que dar dirección y curvatura.

30/1/1986: MICROCHESS 16kB

Se trata del juego de ajedrez adaptado para que ocupe menos de 16kB. El resto de instrucciones van por pantalla.
Todo un desafío para maestros!

REP. ORIENTAL DEL URUGUAY
MEDANOS
SAN PEDRO
CARMELO
ZARATE
CAMPANA
COLONIA DEL SACRAMENTO
LUJAN
BUENOS AIRES
RIO DE LA PLATA
MERCEDES
LA PLATA
CHIVILCOY
LOBOS
CHEL. BRANDSEN
SAN MIGUEL DEL MONTE
CHASCOMUS





Los procedimientos desarrollados por el ingeniero Horacio Reggini pueden ser implementados en la mayoría de las microcomputadoras que operan con decimales.

Implementación del sistema en diferentes equipos



TI-PC: acepta el sistema completo (procedimientos relacionados tanto con la geometría intrínseca de la tortuga tridimensional como con la geometría extrínseca). las Características particulares del sistema Logo implementado en esta máquina —fundamentalmente la orden PLUMADEMEZCLAR— permiten la representación de la tríada en colores.

IBM-PC: acepta el sistema completo (procedimientos relacionados tanto con la geometría intrínseca de la tortuga tridimensional como con la geometría extrínseca). La tríada puede ser representada con el mismo color usado para el dibujo.

Línea Apple: puede operarse el sistema completo, con mayor o menor comodidad según la capacidad de memoria del equipo. La tríada se representa en el mismo color del dibujo.

COMMODORE 64: es simple la implementación del sistema básico (procedimientos relacionados con la geometría intrínseca de la tortuga tridimensional). Como la versión de Logo implementada en esta máquina no acepta la orden VENTANA, la figura debe circunscribirse a las dimensiones de la pantalla. Como tampoco existe la orden PLUMAINVERSA, no es factible realizar fácilmente la representación de la tríada.

SPECTRUM CZ-2000: se puede implementar fácilmente el sistema básico (procedimientos relacionados con la geometría intrínseca de la tortuga tridimensional). No existen restricciones al movimiento de la tortuga, puesto que acepta la orden WINDOW en el procedimiento TRI. Acoplando a la SPECTRUM la impresora térmica pueden obtenerse copias de los dibujos realizados **K64**

TRI-TI-PC 1/2

```
PARA TRI
BP BT OT
VENTANA PPROPORCION 0.55
HACER "F 750
HACER "T11 0
HACER "T12 1
HACER "T13 0
HACER "T21 -1
HACER "T22 0
HACER "T23 0
HACER "T31 0
HACER "T32 0
HACER "T33 1
REEMPLAZAR
HACER "X 0 HACER "Y 0 HACER "Z 0
FIN

PARA GRAFICAR :U :V :W
PXY ON (:F*:U/(:F-:W)) (:F*:V/(:F-:W))
FIN

PARA REEMPLAZAR
HACER "C11 :T11
HACER "C12 :T12
HACER "C13 :T13
HACER "C21 :T21
HACER "C22 :T22
HACER "C23 :T23
HACER "C31 :T31
HACER "C32 :T32
HACER "C33 :T33
FIN

PARA DESOCUPAR
BORRAR ANDAR VIRAR CABECEAR
ROLAR TRI GRAFICAR REEMPLAZAR
BORRAR NOMBRES
BORRAR DESOCUPAR
FIN
```

TRI-TI-PC 2/2

```
PARA ANDAR :D
HACER "X(:x+:D*:C11)
HACER "Y(:Y+:D*:C12)
HACER "Z(:Z+:D*:C13)
GRAFICAR :X :Y :Z
FIN

PARA VIRAR :A
(LOCAL "C "S)
HACER "C COS :A
HACER "S SEN :A
HACER "T11(:C11*:C+:C21*:S)
HACER "T12(:C12*:C+:C22*:S)
HACER "T13(:C13*:C+:C23*:S)
HACER "T21(:C21*:C-:C11*:S)
HACER "T22(:C22*:C-:C12*:S)
HACER "T23(:C23*:C-:C13*:S)
REEMPLAZAR
FIN
PARA CABECEAR :A
(LOCAL "C "S)
HACER "C COS :A
HACER "S SEN :A
HACER "T11(:C11*:C-:C31*:S)
HACER "T12(:C12*:C-:C32*:S)
HACER "T13(:C13*:C-:C33*:S)
HACER "T31(:C31*:C+:C11*:S)
HACER "T32(:C32*:C+:C12*:S)
HACER "T33(:C33*:C+:C13*:S)
REEMPLAZAR
FIN

PARA ROLAR :A
(LOCAL "C "S)
HACER "C COS :A
HACER "S SEN :A
HACER "T21(:C21*:C+:C31*:S)
HACER "T22(:C22*:C+:C32*:S)
HACER "T23(:C23*:C+:C33*:S)
HACER "T31(:C31*:C-:C21*:S)
HACER "T32(:C32*:C-:C22*:S)
HACER "T33(:C33*:C-:C23*:S)
REEMPLAZAR
FIN
```

SUPLEMENTO

DE APLICACIONES

La computadora nos ayuda a resolver muchos problemas y puede sustituírnos en diversas tareas, desde las más elementales de seguridad (prueba de eso es la alarma que logramos poner en marcha con la Drean Commodore 64), hasta escribir por nosotros (con los procesadores que incluimos para las CZ 1000/1500 y TK 83/85, y para la C 64), o hablar con un vocabulario ilimitado (presentamos un sintetizador de voz para CZ 2000, TK 90X y TS 2068), y también podemos dar a la máquina un uso profesional a través de periféricos (con la interfase para CZ 1000/1500, TK 83/85).

CONSTRUYAMOS UNA ALARMA CON LA C-64



Esta vez, una sencilla computadora casera nos prestará un servicio para el cual quizá no haya sido pensada, pero necesario e interesante. Las posibilidades de una alarma controlada por computadora frente a los sistemas electrónicos usuales son muchas. Por ejemplo, se le puede dar prioridad a la apertura de una puerta en especial; sonar la alarma cuando más de dos puertas hayan sido abiertas; o programar una secuencia, sólo conocida por el usuario, frente a la cual el circuito no actúe. En este proyecto, utilizaremos el USER PORT de la C-64, el cual es fácilmente accesible en la parte posterior de la misma. Este port, nos permite transmitir o recibir datos en serie o paralelo, aunque en este último modo con algunas restricciones. En la figura 2, vemos los circuitos de la C-64 que controlan el USER PORT. Estos son dos integrados tipo 6526 CIA, y pueden ser controlados por el usuario mediante pokes adecuados. En la figura 3 podemos ver una tabla en donde se resumen todos los registros del integrado U2. En primer lugar está la dirección del registro en decimal, luego en hexadecimal, y finalmente la función del mismo. En nuestro caso, nos concentraremos en los 4 primeros de la lista. Dado que si bien no es necesario conocer su funcionamiento para llevar a cabo este proyecto, consideramos que es interesante saberlo.

Principio de funcionamiento

Como se puede ver, hay dos ports de entrada/salida: el port A y el port B.

Dado que son de igual funciona-

```
10 POKE 56579,15
15 POKE 56578,PEEK (56578) OR
4
20 FOR B=0 TO 15
30 POKE 56577,B
40 LET A=PEEK (56577) AND 16
45 IF A=16 AND B=15 THEN GO TO
90
50 IF A=16 THEN POKE 56575,PEE
K (56576) AND 251
55 IF A=16 THEN STOP
60 NEXT B
70 GO TO 20
90 FOR C=1 TO 50000: NEXT C
100 GO TO 50
```

miento, nos referiremos sólo a uno de ellos, pero las explicaciones valen para ambos.
En nuestro caso, tendremos accesible el port B.
Veamos el registro de dirección de datos DDRB (DATA DIRECTION REGISTER B).

Figura 1

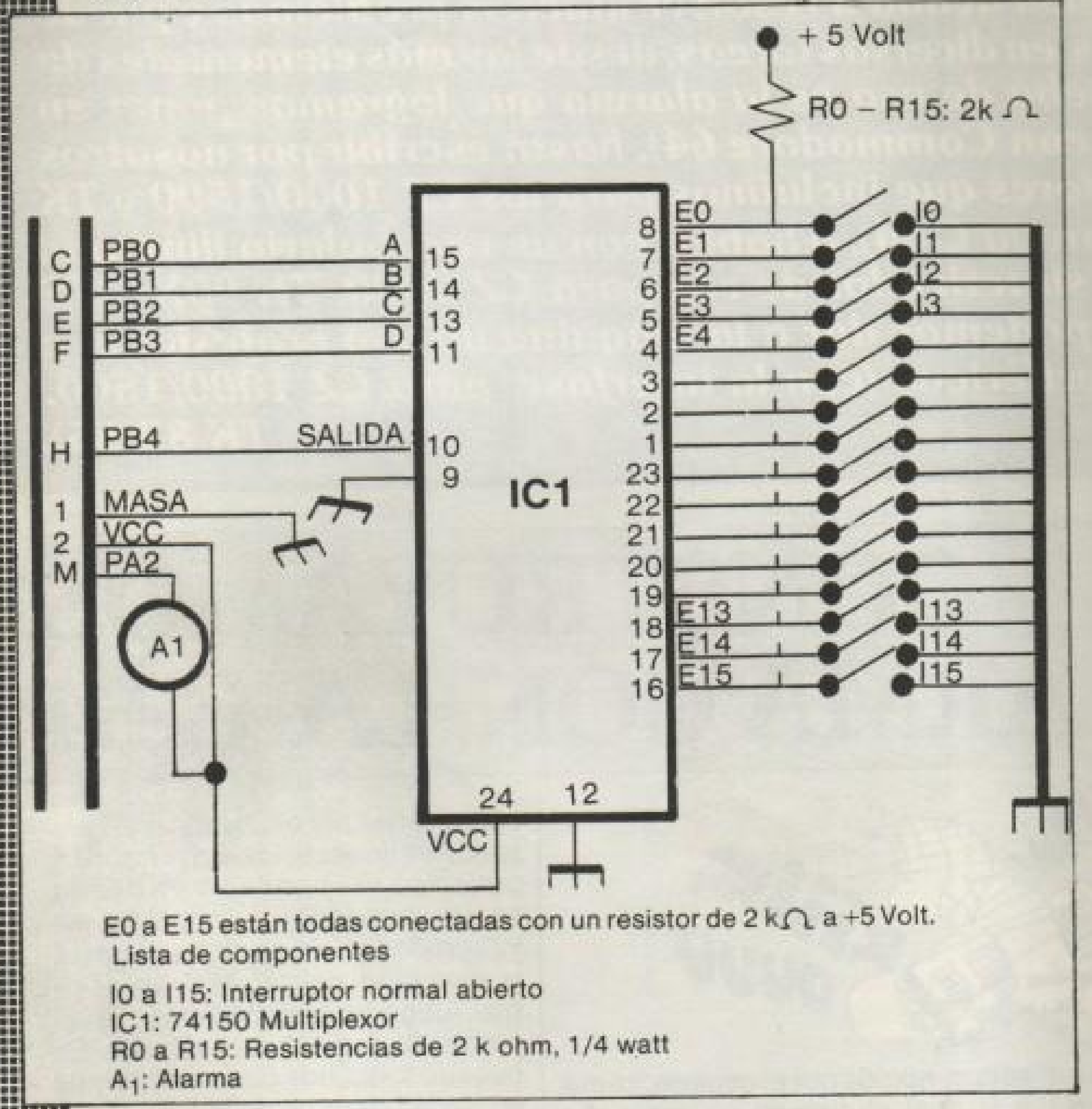
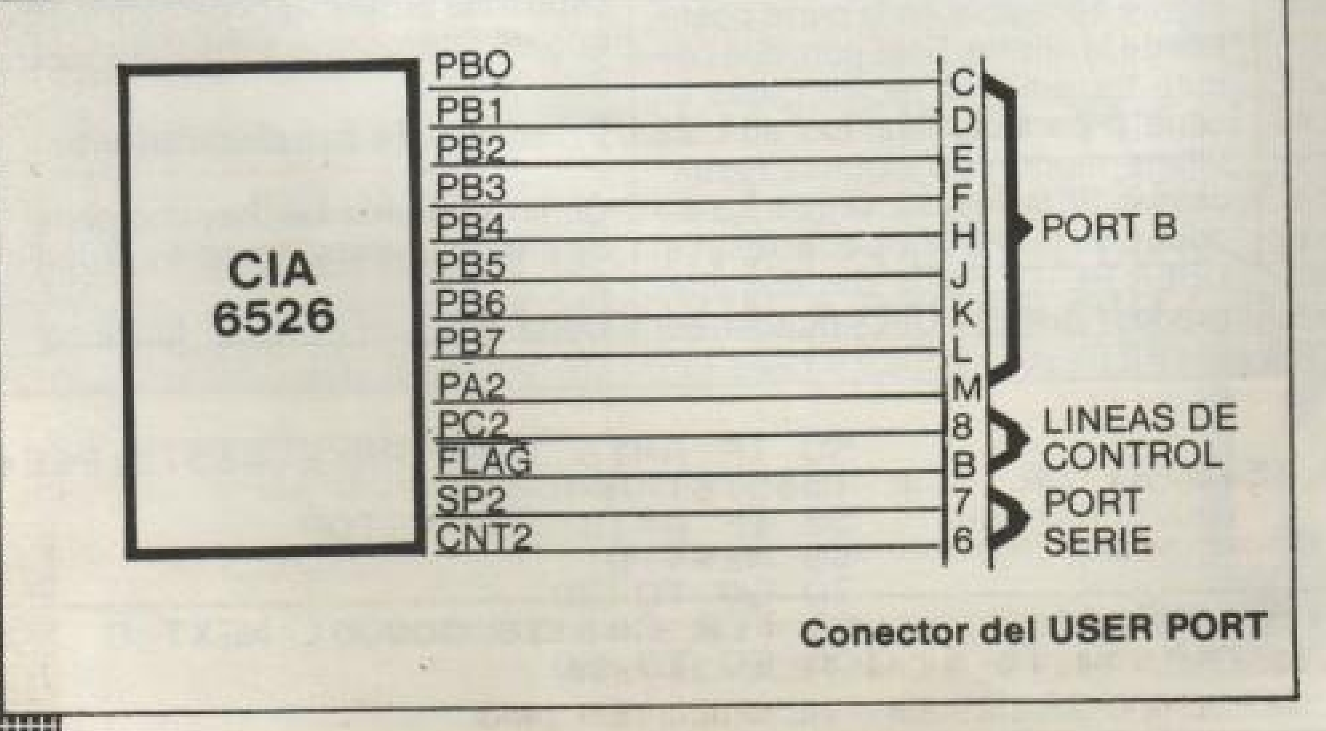


Figura 2



Conector del USER PORT

Como su nombre lo indica, mediante el DDRB podemos controlar la dirección de los datos en cada línea del port B; esto es decidir si la línea PBO será una entrada o salida, lo mismo para el PB1, etcétera.
Asumiendo que el registro es de ocho bits, y que a cada bit le corresponde controlar una línea del port B, ésta será salida si el bit correspondiente es 1 y entrada si es cero.
Por ejemplo, si ejecutamos POKE 56578,255, estamos poniendo todos los bits del registro a 1 y por lo tanto las líneas PB0 a PB7 serán salidas.
Si ahora hacemos POKE 56578,1; como podemos ver en el diagrama, sólo ponemos un bit a 1, el correspondiente a PB, por lo tanto, éste (PB0) será salida y los restantes, entradas.

```
P P P P P P P P
B B B B B B B B
7 6 5 4 3 2 1 0

0 0 0 0 0 0 0 1
```

Ya sabemos cómo hacer para que cada una de las líneas del port B sea una entrada o una salida.
Ahora nos falta ver cómo hacer para poner datos en estas líneas.
Esto se lleva a cabo mediante el Registro de Datos Periféricos PRA (PERIPHAL DATA REGISTER A).
Entonces para poner un dato, digamos el valor 200 en el port B hacemos: POKE 56577,200;
asumiendo que todas las líneas de B fueron inicializadas como salidas.
Si quisiéramos leer un dato presente en el port B, hacemos:
PRINT PEEK 56577.
Al contrario del caso anterior, suponemos que el DDRB fue Pokeado con 0.
Para nuestra alarma, usaremos el port B para controlar un multiplexor.
Esto no es mas que una llave accionada electrónicamente que nos permite ir seleccionando cada uno de los sensores de alarma.
Por ejemplo, cuando en las entradas ABCD del IC1 (fig. 1) tenemos el código binario 0001, estará seleccionando el sensor N1, y la computadora leerá el estado del mismo.
Si en cambio, en las líneas ABCD tenemos el código 0101 (binario que corresponde al 5 en decimal, la computadora leerá el estado del sensor número 5.
Entonces, si en las líneas ABCD ponemos todos los códigos desde el 0000 al 1111, iremos leyendo el estado de todos los sensores, desde el número 0 al 15, y determinar si existe una condición de alarma o no.
En la figura 1 podemos ver el esquema eléctrico completo de la alarma.
Por sensores utilizaremos inte-

ruptores de cualquier tipo, basta con que se puedan adaptar a nuestras necesidades, como ser colocados en el marco de una puerta y accionarse al abrirse la misma.

Si una puerta es abierta, los contactos del interruptor correspondientes se cerrarán causando que aparezca un potencial de 0 volt en la entrada correspondiente de IC1.

Mediante un sencillo programa, se hace un "barrido" de las 16 entradas de IC1 generando un contador binario en las líneas PB0 a PB3, que están eléctricamente conectadas a ABCD.

Luego de cada conteo, la salida de IC1 (terminal 10) es sensada leyendo el estado de PB4.

Si en PB4 hay un potencial alto (equivale a el bit en 1) significa que la alarma no ha sido disparada.

De lo contrario (el bit a 0) alguno de los interruptores ha sido activado y la alarma debe dispararse.

Para ello, se genera un bajo potencial en PA2 causando de este modo que suene la alarma.

Explicación del programa

LINEA 10: pokea o escribe el valor 15 (00001111) en el registro de dirección de datos del port B. Esto hace que las líneas PB0 a PB3 sean salidas, y PB4 a PB7 entradas. Las líneas PB0 a PB3 son utilizadas para generar una cuenta en binario (de 0000 a 1111) en las líneas ABCD del multiplexor. Este nos dará una salida en el terminal 10 que representa el estado del interruptor seleccionado por el contador binario.

LINEA 15: pokea o escribe el registro de dirección de datos tan sólo cambiando el bit correspondiente al PA2 y manteniendo los otros bits inalterados.

Esto es necesario porque estas otras líneas controlan el bus serie y no deben ser modificadas.

LINEA 20: es el loop que genera cuenta de 0 a 15. Esta seleccion rá una de las 16 entradas del mu tiplexor cuando sea escrita en el r gistro de datos periféricos del po B en la línea 30.

LINEA 30: pokea o escribe cuenta binaria en PRB.

LINEA 40: lee el PB y se ejecu un AND 16 para que el único bit ir portante sea el bit 4, es decir línea PB4 que representa el esta do del interruptor seleccionado. el valor leído es 16, esto indicar que el interruptor correspondien te fue cerrado, y la alarma debe a tivarse.

LINEA 45: nos da un retardo en disparo de la alarma en caso d que la puerta abierta haya sido un escogida de antemano. Para nues tro ejemplo será la controlada po el interruptor número 15, pero est puede modificarse cambiando valor que se le asigna a la variabl B. Por ejemplo, podemos poner retardo en el interruptor corres pondiente a la puerta principal, po la cual nosotros entraremos e ire mos a desconectar la alarma ante de que se active.

LINEA 50: activa la alarma ponier do PA2 en 0. Esto se debe hacer si modificar las otras líneas del port de ahí que se recurra al AND 251

LINEA 55: finaliza el programa s la alarma es activada.

LINEA 60: incrementa el contado que selecciona la próxima entrada del multiplexor.

LINEA 70: salta a la línea 20, lue go de que la cuenta alcanza 15, re seteando el contador a 0 y comen zando el barrido nuevamente.

LINEA 90: genera un retardo para los casos de entrada normal. Este retardo puede ser variado cambian do el valor del loop. **K64**

Figura 3

DECIMAL	HEXA		REGISTRO
56576	DD00	PRA	Registro de datos periféricos A
56577	DD01	PRB	Registro de datos periféricos B
56578	DD02	DDRA	Registro de dirección de datos A
56579	DD03	DDRB	Registro de dirección de datos B
56580	DD04	TA	Registro del timer A (bajo)
56581	DD05	TA	Registro del timer A (alto)
56582	DD06	TB	Registro del timer B (bajo)
56583	DD07	TB	Registro del timer B (alto)
56584	DD08	TOD	Reg. de hora diaria (1/10 seg.)
56585	DD09	TOD	Reg. de hora diaria (segundos)
56586	DD0A	TOD	Reg. de hora diaria (minutos)
56587	DD0B	TOD	Reg. de hora diaria (horas)
56588	DD0C	SDR	Registro de datos serie
56589	DD0D	ICR	Registro de control de interrup.
56590	DD0E	CRA	Registro de control A
56591	DD0F	CRB	Registro de control B

PROCESADOR DE TEXTOS



COMP.: CZ 1000/1500, TK 83/85
CONF.: IMPRESORA 16 K RAM



El programa que les ofrecemos a continuación nos permite conectar a nuestra computadora una sencilla impresora térmica para utilizarla como procesador de textos.
Entre las características del programa se cuentan las siguientes:
-Efectúa automáticamente justificación a derecha cuando una línea excede los 32 caracteres.
-Permite alterar el texto, definir un nuevo párrafo e indicar la finalización del mismo.
-Realiza una impresión automática del texto una vez que se llena la pantalla; hecho esto, lo borra y espera la introducción de más texto.
-Para aquellos que no posean una impresora, podrán de todos modos correr este programa y verlo en funcionamiento mediante una sencilla modificación del mismo.

Comandos del sistema

Los comandos del sistema son pocos y sencillos de utilizar:
".": indica el final de un párrafo. De este modo, automáticamente bajamos de renglón para tipear una nueva línea.
"/": Indica el final del texto. Al tipear este caracter ocurren las siguientes cosas: 1) Se realiza una impresión automática del texto existente hasta el momento.
2) Borra la pantalla y espera que más texto sea introducido.
"0": Entra al modo de edición de textos.
"NEWLINE": Equivale a espacio, y luego de alterar el texto; presionando SHIFT y N/L permite agregar más texto al ya impreso.

Modo de edición

Al entrar al modo de edición, presionando 0, podrá verse la primera letra del texto en modo inverso. Mediante las teclas 5, 6, 7 y 8 (manteniéndolas apretadas) se podrá mover el cursor inverso en la dirección señalada por las flechas. Cuando el cursor esté sobre la letra que se desee borrar, basta con tipear la nueva letra sobre la anterior.
Para aquellos que no posean impresora, pero deseen ver cómo trabaja el programa deben cambiar las siguientes líneas:

330 PRINT AT 0,0;"COPIANDO"
331 INPUT I $
en vez de:
330 COPY
sin línea 331 **K64**

```
   1 REM PROCESADOR DE TEXTOS
   5 DIM S(32)
  10 LET C$=""
  20 LET L=0
  30 PRINT AT L,0;C$
  40 INPUT B$
  50 IF B$="." THEN GOTO 275
  60 IF B$="/" THEN GOTO 330
  70 IF B$="0" THEN GOTO 360
  75 IF B$="S" THEN STOP
  80 LET C$=C$+B$+" "
  90 IF LEN C$<32 THEN GOTO 30
  91 FOR I=32 TO 1 STEP -1
  92 IF C$(I)<>" " THEN NEXT I
 100 LET A$=(C$( TO I-1)+"
                             ")(1 T
O 32)
 110 LET C$=C$(I+1 TO )
 120 LET X=0
 130 FOR I=1 TO 32
 140 IF A$(I)<>" " THEN GOTO 170
 150 LET X=X+1
 160 LET S(X)=I
 170 NEXT I
 180 IF A$(32)<>" " THEN GOTO 30
0
 185 PRINT AT L,0;A$
 190 LET A=INT (RND*X+1)
 195 IF S(A)>32 THEN GOTO 190
 200 FOR I=32 TO S(A)+1 STEP -1
 210 LET A$(I)=A$(I-1)
 220 NEXT I
 230 FOR I=A TO X
 240 LET S(I)=S(I)+1
 245 NEXT I
 250 GOTO 180
 275 LET C$=""
 280 LET L=L+1
 290 IF L=21 THEN GOTO 330
 300 LET L=L+1
 310 IF L-21 THEN GOTO 30
 330 PRINT AT 0,0;"COPIANDO"
 331 INPUT I$
 340 CLS
 350 GOTO 10
 360 LET Y=0
 370 LET X=0
 380 GOSUB 1000
 400 IF INKEY$="" THEN GOTO 400
 410 LET A=CODE INKEY$
 420 IF A=118 THEN LET A=0
 425 IF A<64 OR A>127 THEN GOTO
500
 427 GOSUB 1000
 430 IF A=112 THEN LET Y=Y-1
 440 IF A=113 THEN LET Y=Y+1
 450 IF A=114 THEN LET X=X-1
 460 IF A=115 THEN LET X=X+1
 470 IF A-121 THEN GOSUB 1000
 480 IF A=121 THEN GOTO 40
 490 GOTO 390
 500 PRINT AT Y,X;CHR$ A
 510 LET X=X+1
 520 IF X<32 THEN GOTO 550
 530 LET X=0
 540 LET Y=Y+1
 560 GOSUB 1000
1000 PRINT AT Y,X;
1010 LET U=PEEK (PEEK 16398+256*
PEEK 16399)+128
1030 IF U>255 THEN LET U=U-256
1040 PRINT CHR$ U
1050 RETURN
```

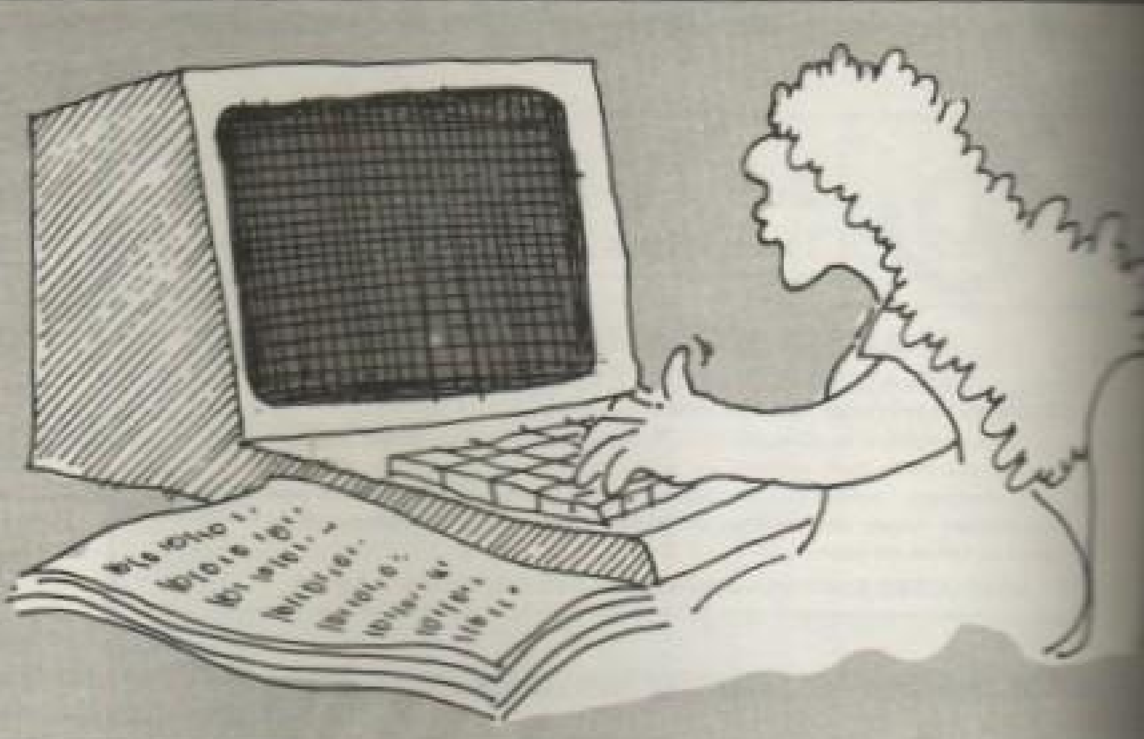
C U R S O S

PROCESA TEXTO



COMP.: DREAN COMMODORE 64
CLAS.: UTI

Este programa es un procesador de texto que posibilita utilizar la Commodore como máquina de escribir, y archivar en caset o imprimir con la impresora el texto deseado. Nos permite cargar un texto en una cinta, copiarlo por impresora, corregirlo.



```
1 PL=66:LS=1:TM=5:BM=5:PN=1:PN$="V":CC=1
2 POKE808,225
3 GOTO18
4 PRINT"@ ■■";:FORX=0TO55:GETX$:IFX$=""T
HENNEXT:PRINT" ■";:FORX=0TO45:GETX$:IFX
$=""THENNEXT:GOTO4
5 IF(X$=CHR$(34))THENX$="'"
6 X=ASC(X$):X0=VAL(X$):PRINT" ■";:RETURN
7 GOSUB4:IF(X=13)OR(X=20)OR(X=133)THENRE
TURN
8 IF(X=95)THENGOSUB154
9 IF(X<32)OR((X>128)AND(X<160))THEN7
10 RETURN
11 GOSUB7:IF(X=13)OR(X=133)THENRETURN
12 IF(NU=0)AND((X>57)OR(X<48))THENGOTO12
13 IFY=0ANDX=20THENGOTO11
14 IFX=20THENV$=LEFT$(V$,LEN(V$)-1):Y=V-
1:PRINTX$;:GOTO11
15 IFY=LTHENX$="":GOTO11
16 Y=Y+1:V$=V$+X$:PRINTX$;:IFLEN(V$)=TW
-5THENGOSUB149
17 GOTO11
18 POKE53280,0:POKE53281,0:E=31744:PRINT"
19 PRINTCHR$(14)CHR$(8)"■■■■■■
20 PRINT"@ ROCESATEXTO  ■"
21 PRINT"■V \IKE -ICHTER -- \OD BY IIM
ΓMOSKO"
22 PRINT"■                    \OD BY -. \
LIPPERT"
23 PRINT"■\OD & TRADUCIDO FOR -.-.TINEIR
O Y LEONE"
24 FORX=1TO5000:NEXTX:PRINT"■"
25 DIMT$(500),TB(25)
26 PRINTCHR$(14)
27 PRINTCHR$(147):PRINTSPC(9);"■■■■■■■
■■■■■■":PRINT:PRINT
28 PRINT:PRINT"■/■■■/ ?     ■■EER LO
ESCRITO"
29 PRINTSPC(15);"■■DITAR LO ESCRITO"
16 Y=Y+1:V$=V$+X$:PRINTX$;:IFLEN(V$)=TW-
5THENGOSUB149
17 GOTO11
18 POKE53280,0:POKE53281,0:E=31744:PRINT

19 PRINTCHR$(14)CHR$(8)"■■■■■■
20 PRINT"@ ROCESATEXTO  ■"
21 PRINT"■V \IKE -ICHTER -- \OD BY IIM
ΓMOSKO"
22 PRINT"■                    \OD BY -. \
LIPPERT"
23 PRINT"■\OD & TRADUCIDO FOR -.-.TINEIR
O Y LEONE"
24 FORX=1TO5000:NEXTX:PRINT"■"
25 DIMT$(500),TB(25)
26 PRINTCHR$(14)
27 PRINTCHR$(147):PRINTSPC(9);"■■■■■■■
■■■■■■":PRINT:PRINT
28 PRINT:PRINT"■/■■■/ ?     ■■EER LO
ESCRITO"
29 PRINTSPC(15);"■■DITAR LO ESCRITO"
30 PRINTSPC(15);"■■GREGAR ALGO"
31 PRINTSPC(15);"■■NGRESAR DE CINTA"
32 PRINTSPC(15);"■■RABAR EN CINTA"
33 PRINTSPC(15);"■■OPIA IMPRESA"
34 PRINTSPC(15);"■■OSICIONAR TABULADORE
S"
35 PRINTSPC(15);"■■ABER MEMORIA DISPONI
BLE"
36 PRINTSPC(15);"■■OLVER AL IASIC"
37 PRINT"               O ■■EXTO NUEVO"
38 PRINT"■■■ETERMINE SU OPCION   ";:GOS
UB4
39 IFX$="L"GOTO50
40 IFX$="A"THENGOSUB159:IFK<>999THENGOTO
89
41 IFX$="P"GOTO95
42 IFX$="E"GOTO58
43 IFX$="T"GOTO88
44 IFX$="I"GOTO102
45 IFX$="G"GOTO112
46 IFX$="C"GOTO164
47 IFX$="S"GOTO124
48 IFX$="V"THENPRINT"■■■ I ■STA UD.■"
■■■";:GOSUB4:IFX$="V"THENSYS64738
49 K=0:GOTO27
50 I=0:PRINTCHR$(147)
51 IFT$(I)="£"GOTO54
52 IF(I>0ANDI/12=INT(I/12))THENGOSUB55
53 PRINTT$(I):I=I+1:GOTO51
54 GOSUB55:GOTO27
55 PRINT"■■■■SO ES TODO EN ESTA PAGINA
.":PRINT"■ULSE '■■■' P/CONTINUAR"
56 GETX$:IFX$=""GOTO56
57 PRINTCHR$(147):RETURN
58 GOTO67
59 PRINT"■ARA VOLVER A \■/ PULSE '-1'.
"
60 PRINT"■ARA BUSCAR UNA PALABRA PULSE
'-3'."
61 PRINT"■ARA IR A LINEA ANTERIOR PULSE
'-5'."
62 PRINT"■ARA IR A PROXIMA LINEA PULSE
'-7'."
63 PRINT"■ARA BORRAR LA LINEA TIPEE ■."
64 PRINT"■ARA INSERTAR NUEVA LINEA TIPE
E '↑'."
65 PRINT"■ARA CORREGIR LA LINEA TIPEE ■
."
66 PRINT"■PULSE <■■■> CUANDO EL TEXTO
 ESTE Γ./.■":RETURN
67 SR$="********************************
*********":NL$="■NGRESE NUEVA LINEA ■"
68 K=0
69 GOSUB59
70 IFT$(K)=""THENPRINTSR$:PRINT"■■■ ×
■■■) ";:GOSUB4:GOTO74
71 IFK=500THENPRINTSR$"■■Γ ■■ I■■Γ -
■ULSE UNA TECLA ";:GOSUB4:GOTO27
72 IFT$(K)="£"GOTO27
73 PRINTSR$:PRINTT$(K)" ";:GOSUB4
74 IFX$="■"GOTO82
75 IFX$="↑"GOTO84
76 IFX$="■"GOTO86
77 IFX$=CHR$(133)GOTO27
78 IFX$=CHR$(134)THENGOSUB223:GOTO69
79 IF(X$=CHR$(135))AND(K<>0)THEN K=K-1:GO
TO69
80 IFX$=CHR$(136)THENK=K+1:GOTO69
81 GOSUB4:GOTO74
82 FORL=KTO499:T$(L)=T$(L+1):NEXTL
83 GOTO69
84 PRINT:FORL=500TOK+1STEP-1
85 T$(L)=T$(L-1):NEXTL:PRINT:PRINTNL$:L=
TW:Y=0:V$="":GOSUB11:T$(K)=V$:K=K+1:GOTO
69
86 PRINT:PRINTSR$:PRINT"■  "T$(K)"T":IFL
EN(T$(K))>38THENPRINT"III"
87 INPUTT$(K):K=K+1:GOTO69
88 K=0:PRINTCHR$(147):GOSUB127
89 IFTW=0THENGOSUB127
90 L=TW:Y=0:V$="":GOSUB11:T$(K)=V$
91 FR=FRE(0)-(FRE(0)<0)*65536:IFFR<1000G
OTO94
92 IFK=499THENGOTO94
93 IFX$<>CHR$(133)THENK=K+1:PRINT:GOTO■■
94 K=K+1:T$(K)="£":GOTO27
95 PRINTCHR$(147):PRINT" ".-IJE TABULAD
ORES":PRINT:PRINT
96 PRINT"L TABULADOR ES LA TECLA '■'."
PRINT"(■NGULO SUP.IZQUIERDO.)■"
97 PRINT"■ULSE '-1' PARA VOLVER AL \■■
■■":I=1:TT=0
98 PRINT"■UANTOS ESPACIOS A TAB"I"' ?
 ■■■■■■";:Y=0:V$="":L=2:NU=9
99 GOSUB11:NU=0:TB(I)=VAL(V$)-1+TT:IF■■
33THENGOTO27
100 TT=TB(I):IFTT>75THENGOTO27
101 I=I+1:PRINT"T":GOTO98
102 PRINT"■■LEYENDO DE CINTA.■■"
103 OPEN1,1,0:I=0:PRINT"■"
104 INPUT#1,LM:INPUT#1,RM:INPUT#1,TW:INP
UT#1,PL:INPUT#1,LS
105 INPUT#1,TM:INPUT#1,BM:INPUT#1,PN:INP
UT#1,PN$
106 INPUT#1,X$
107 IFX$="£"GOTO110
108 T$(I)=X$
109 I=I+1:GOTO106
110 T$(I)="£":CLOSE1:K=I
111 GOTO27
112 PRINT"■■RABANDO EN CINTA.■■"
113 INPUT"/OMBRE DEL ARCHIVO?";X$
114 OPEN1,1,1,X$:I=0:PRINT"■"
115 PRINT#1,LM:PRINT#1,RM:PRINT#1,TW:PRI
NT#1,PL:PRINT#1,LS
116 PRINT#1,TM:PRINT#1,BM:PRINT#1,PN:PRI
NT#1,PN$
117 IFT$(I)="£"THENGOTO121
118 X$=CHR$(34)+T$(I)+CHR$(34):GOTO119
119 PRINT#1,X$
120 I=I+1:GOTO117
121 PRINT#1,"£"
122 M=0
123 CLOSE1:GOTO27
124 PRINT"■ALTO":FR=FRE(0)-(FRE(0)<0)*■■
536:PRINT"■■■■■■■        \EMORIA LIBRE:"FR
125 PRINT"■■■ULSE UNA TECLA PARA RETORN
AR AL PROGRAMA.";:GOSUB4
126 GOTO27
127 PRINT"■■■   .  STE PROGRAMA OPERA CO
MO UNA MA-"
128 PRINT"■QUINA DE ESCRIBIR.  ETERMINE
LOS MARGE-"
129 PRINT"■NES DE AMBOS LADOS. ENTONCES
PODRA PRO-"
130 PRINT"■CEDER A ESCRIBIR ENTRE ELLOS
SU TEXTO. "
131 PRINT"■-UANDO ESTE PROXIMO AL LIMITE
 DE LA LI-"
132 PRINT"■NEA. SE OIRA UN BEEP DE ALERT
A. "
133 PRINT"■LA TECLA DE BORRADO PUEDE SER
USADA CO-"
134 PRINT"■MO GOMA, Y LA TECLA '←' DE TA
BULADOR."
135 PRINT"■■■/L■"  //■ I■L■ "■Γ/■■■
.";:GOSUB4
136 PRINT"■■ARA CENTRAR UNA LINEA USE
EL SIMBOLO"
137 PRINT"■ _Γ  (■I,■-L■ ■) COMO PRIMER
CARACTER."
138 PRINT"■I DESEA PASAR A OTRA PAGINA
USE LA TE"
139 PRINT"■LA '↑' COMO EL UNICO CARACT
```

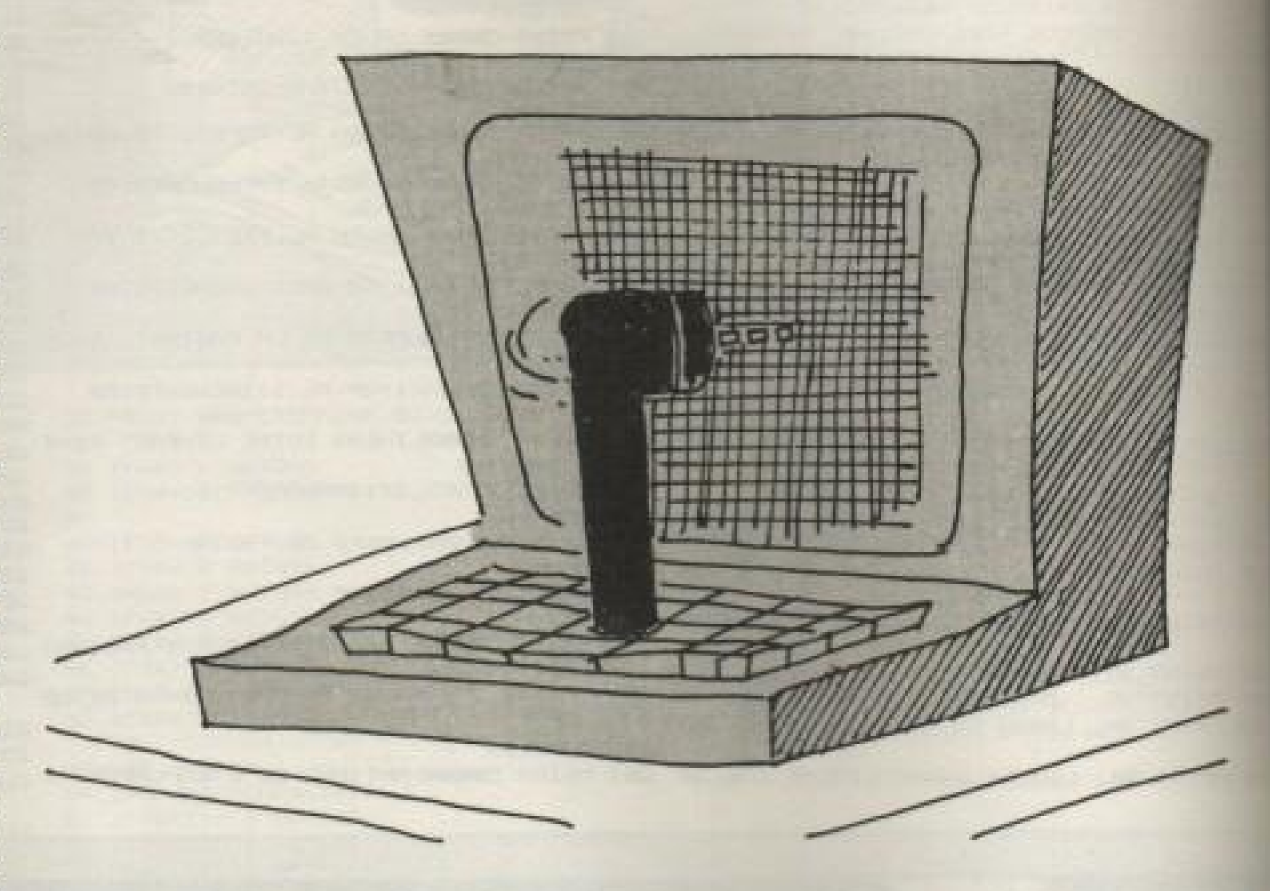
```
R DE LINEA"
140 PRINT"N/NA VEZ QUE QUE HALLA FINALIZ
ADO CON SU"
141 PRINT"NTEXTO, PULSE "-1" PARA VOLVER
AL ."
142 PRINT"N/NA VEZ EN EL MISMO USTED POD
RA DETERMI"
143 PRINT"NAR CUALQUIER TAREA A SEGUIR.
"
144 PRINT"N-IJE MARGEN IZQUIERDO.(3-70)
";:INPUTLM
145 IF(LM>70)OR(LM<3)THENGOTO144
146 PRINT"N-IJE MARGEN DERECHO.(3-70):";
:INPUTRM:TW=80-(RM+LM)
147 IF(TW<74)OR(RM<3)THENGOTO146
148 PRINT"PUEDE COMENZAR:":RETURN
149 S4=54272:POKES4+5,9:POKES4+6,0:POKES
4+24,15
150 POKES4+1,100:POKES4,250:POKES4+4,33
151 FORE4=0TO50:NEXT:POKES4+24,0
152 POKES4+1,00:POKES4,0:POKES4+4,00
153 RETURN
154 FORTA=0TO25:IFTB(TA)>YTHEN156
155 NEXTTA:X=0:RETURN
156 N0=Y:N2=Y:N1=TB(TA):TA=25:NEXT
157 FORN2=N2TON1-2:Y$=Y$+" ":Y=Y+1:PRINT
" ";:NEXT
158 X$=" ":X=32:RETURN
159 K=0
160 IFT$(K)="£"THENK=K-1:PRINT"":RETURN
161 IFK=500THENPRINT:PRINT" - ULSE UNA TECLA: ";
162 IFK=500THENGOSUB4:K=999:RETURN
163 K=K+1:GOTO160
164 C1=0
165 PRINT"
166 PRINT"1. MARGEN IZQUIERDO"TAB(30)LM
167 PRINT"2. MARGEN AL TOPE"TAB(30)TM
168 PRINT"3. MARGEN AL PIE"TAB(30)BM
169 PRINT"4. LARGO DE LA PAGINA"TAB(30)
PL
170 PRINT"5. LINEAS ENTRE LINEAS"TAB(30
)LS
171 PRINT"6. NUMERO DE PAGINA"TAB(30)PN
172 PRINT"7. IMPRIME NUMERO DE PAGINA"
TAB(30)" "PN$
173 PRINT"8. CANTIDAD DE EJEMPLARES"TAB
(30);CC
174 PRINT"PULSE "-1" PARA PROCEDER A I
MPRIMIR O"
175 PRINT"UN NUMERO (1-8) PARA CUALQUIE
R OPCION.";
176 GOSUB4
177 IFX=13GOTO204
178 IFX<>0THENONXGOSUB180,183,186,189,1
92,195,198,201
179 GOTO165
180 PRINT"MARGEN IZQUIERDO? (3-70):
";:INPUTLM
181 IF(LM>70)OR(LM<3)THENGOTO180
182 RETURN
183 PRINT"MARGEN AL TOPE?. (5-60):
";:INPUTTM
184 IF(PL-TM-BM<1)OR(TM<5)THENGOTO183
185 RETURN
186 PRINT"MARGEN AL PIE?.. (5-60):
";:INPUTBM
187 IF(PL-TM-BM<1)OR(BM<5)THENGOTO186
188 RETURN
189 PRINT"LARGO DE LA PAGINA? (1-26
4): ";:INPUTPL
190 IF(PL-TM-BM<1)OR(PL<1)THENGOTO189
191 RETURN
192 PRINT"LINEAS ENTRE LINEAS? (1-4
): ";:INPUTLS
193 IF(LS>4)OR(LS<1)THENGOTO192
194 RETURN
195 PRINT"NUMERO DE PAGINA? (1-?):
";:INPUTPN
196 IFPN<1THENGOTO195
197 RETURN
198 PRINT"IMPRIME NUMERO DE PAGINA?
(S/N) ";:GOSUB4
199 IF(X$="S")OR(X$="N")THENPN$=X$:RETUR
N
200 GOTO198
201 PRINT"CANTIDAD DE EJEMPLARES? (
1-?): ";:INPUTCC
202 IFCC<1THENGOTO201
203 RETURN
204 PRINT"VERIFIQUE QUE SU IMPRESORA
ESTE LISTA,":K=0:P3=PL-TM-BM:P1=0
205 PRINT"Y QUE EL PAPEL ESTE ALINEADO.

206 PRINT"
";:GOSUB4:P2=PN:OPEN4,4,7
207 FORI=1TOTM:PRINT#4:NEXT
208 IFLEFT$(T$(K),1)="_"THENGOTO215
209 IFT$(K)="↑"THENGOSUB220:GOSUB217:K=K
+1:GOTO207
210 IFT$(K)="£"THENGOSUB221
211 PRINT#4,""SPC(LM)T$(K):P1=P1+1
212 IFLS<>1THENFORI=2TOLS:PRINT#4:P1=P1+
1:NEXT
213 IFP1>P3-1THENGOSUB217:GOTO207
214 K=K+1:GOTO208
215 PRINT#4,""SPC(LM)SPC(INT((TW-LEN(T$(
K))-2)/2))RIGHT$(T$(K),LEN(T$(K))-1)
216 P1=P1+1:GOTO212
217 PRINT#4:IFPN$="S"THENPRINT#4,""SPC(L
M)SPC(INT((TW-7)/2))"HOJA"P2:GOTO219
218 PRINT#4
219 P2=P2+1:FORI=3TOBM+P3-P1:PRINT#4:NEX
T:P1=0:RETURN
220 P5=P1:FORP1=P5TOP3-1:PRINT#4:NEXT:RE
TURN
221 GOSUB220:GOSUB217:C1=C1+1:IFC1=CCTHE
NCLOSE4:GOTO26
222 P1=0:P2=PN:K=0:GOTO207
223 PRINT"QUE PALABRA O FRASE DESEA
ENCONTRAR?":PRINT:G=K
224 Y=0:Y$="":L=TW:GOSUB11:PRINT:PRINT:P
RINT"   BUSCANDO: "Y$
225 K=K+1:IFT$(K)="£"THENGOTO231
226 IFLEN(T$(K))<LEN(Y$)THENGOTO225
227 IFT$(K)=Y$THENRETURN
228 FORV=1TOLEN(T$(K))-LEN(Y$)+1
229 IFMID$(T$(K),V,LEN(Y$))=Y$THENRETURN

230 NEXT:GOTO225
231 PRINT"   - PULSE UNA TE
CLA.";:GOSUB4:K=G:RETURN
```

INTERFASE DE CONTROL PARA CZ 1000/1500, TK 83/85

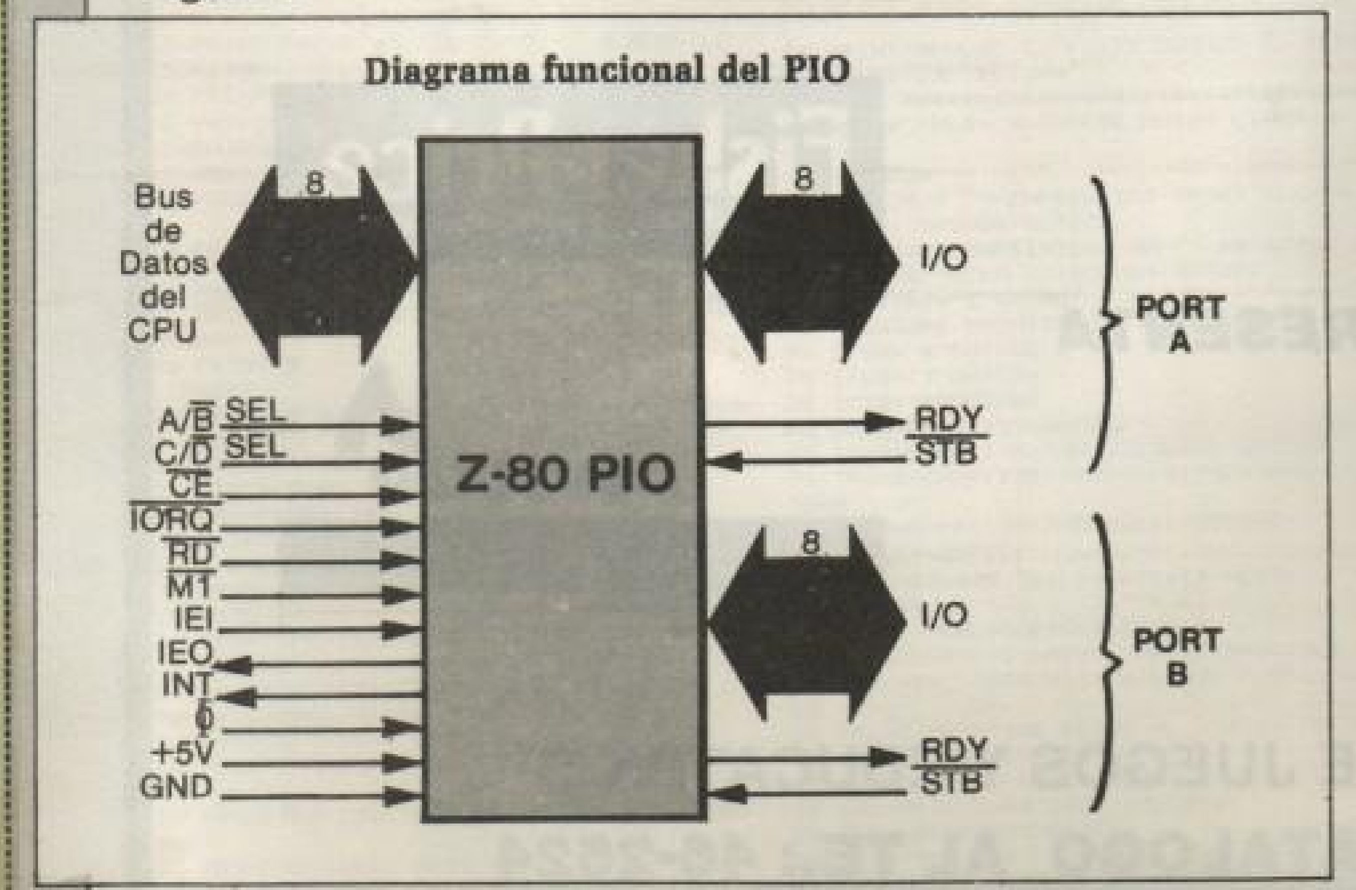
Casi toda persona que posea una microcomputadora habrá sido alguna vez interrogada acerca de qué cosas prácticas se pueden hacer con la misma.
Una de las aplicaciones más obvias es la de controlar dispositivos externos; pero, desgraciadamente, muy pocos microprocesadores o CPUs son diseñados para hacer esto directamente.
La mayoría de los fabricantes de microprocesadores ofrecen dispositivos (comúnmente circuitos integrados) llamados controladores de periféricos y están hechos para funcionar específicamente con una CPU. Estos controladores simplifican en gran forma la tarea de interfacear la microcomputadora con dispositivos periféricos, como ser disketeras, impresoras o modems.
Afortunadamente, los ingenieros de Sinclair Research diseñaron su microcomputador en base al CPU Z-80 para el cual hay muy buenos controladores de periféricos. Uno de ellos, el Z-80 PIO, puede ser utilizado para construir una sencilla interfase paralelo para los microcomputadores del tipo ZX 81 (CZ 1000/1500, TK 83/85).



El Z-80 PIO

El Z-80 PIO (Pheripal Input Output) es un circuito integrado de 40 terminales diseñado para servir de interfase paralelo entre la CPU Z-80 y dispositivos periféricos. La comunicación entre el PIO y la CPU se produce conectando las líneas de datos del PIO directamente a las de la CPU. El PIO es un dispositivo de 2 ports. Esto significa que puede enviar y/o recibir dos sets de 8 bits de datos en paralelo.
Las líneas de control del PIO seleccionan uno de los dos ports (B/A SEL), habilitan el PIO (CE), y permiten que el PIO diferencia las palabras de control de las de datos (C/D SEL).
Las otras tres líneas de control (M1, IORQ, RD) aseguran un timing correcto durante las operaciones de entrada/salida con la CPU. Las barras sobre el nombre de las señales indican que su estado activo es el bajo.
Cada port tiene dos líneas de control (RDY y STB) que son utilizadas (no siempre) para sincronizar la entrada o salida de datos. En otras palabras, una de estas líneas, la RDY, debe ser activada para decirle a un dispositivo que está enviando datos al PIO algo así: "No manden más datos ahora. Todavía no estoy listo... O.K. ahora estoy listo. Manden datos ahora". El dispositivo debe responder activando la línea STB, diciendo: "O.K. acá van los datos. Tómalos ahora así puedo hacer otras cosas". Mediante el uso de estas dos líneas (lla-

Figura 1



Las máquinas diseñadas en base al microprocesador Z-80 disponen de buenos controladores de periféricos. Uno de ellos, el Z-80 PIO puede ser utilizado para el desarrollo que se explica en estas páginas.

madas en inglés "handshake") la transmisión de datos entre el dispositivo periférico y el PIO se realiza en forma ordenada y eficiente. El PIO contiene un número de registros internos que son utilizados para controlar su operación.

El más importante es un registro mediante el cual se pueden seleccionar diferentes modos de operación para ambos ports.

El PIO puede ser operado en uno de cuatro modos, siendo estos designados modo 0 a 3. Modo 0 es el modo de salida: las ocho líneas en el port designado son utilizadas como salidas a algún dispositivo externo. En modo 1, el modo de entrada, todas las líneas del port serán entradas, recibiendo datos de algún dispositivo externo. Modo 2 es el modo bidireccional y está restringido al port A. En este modo, las líneas de control del port B, junto con las del port A, son utilizadas para controlar el flujo de datos en ambas direcciones del port A. Modo 3, el modo de control, es un híbrido entre el modo de entrada y el de salida; cualquier línea del port especificado puede ser definida como entrada o salida.

El modo de control difiere del modo bidireccional en que una vez que una línea es designada como entrada o salida, se mantiene en esa condición y es necesario reprogramar para alterar la dirección de transferencia de datos en esa línea. Las líneas de control no son utilizadas en modo 3. Una explicación más detallada de los modos de operación del PIO puede ser encontrada en el libro Z-80 PIO Z-80A PIO TECHMICAL MANUAL, Zilog, 1977.

Otros registros internos del PIO son utilizados para almacenar el vector de interrupciones, una particularidad que distingue al Z-80 PIO.

Ahora veremos como construir y programar una interfase paralelo para las CZ 1000/1500 y TK 83/85 utilizando este PIO mediante control sin interrupciones ni "handshake".

Construcción

La figura 2 nos muestra el circuito

eléctrico de la interfase paralelo. El port B es utilizado como entrada para leer el estado de ocho interruptores (I1-I8), mientras que el port A, es configurado como salida para manejar ocho diodos emisores de luz (LEDs). Se utilizan compuertas inversoras para amplificar la salida de cada línea del port de salida. La máxima corriente capaz de entregar el PIO es de alrededor de 1,5 miliamperes, lo cual no es suficiente para iluminar un led pero si para manejar una entrada TTL, o cuatro entradas tipo LS TTL. Los pulsos de reloj invertidos se hallan presentes en el terminal 6B, del conector trasero de la computadora. Esta señal debe invertirse nuevamente antes de ser suministrada al PIO. Las líneas de control, STB y RDY de cada port, no son utilizadas. No son necesarias porque se seleccionará el modo 3 de trabajo. Dado que esta aplicación no necesita interrupciones, el terminal IEI (Interrupt Enable In) será conectado a +v, y el IEO (Interrupt Enable Out) no es conectado.

Las señales presentes en el conector trasero de la máquina pueden ser utilizadas mediante un conector de borde apropiado.

Este debe ser de 50 terminales, con un espacio entre contactos de 0,1 pulgada. Probablemente, el conector de borde deba ser modificado pues los que venden en los comercios tienen sus extremos cerrados, y nosotros necesitamos que estén abiertos.

Esto se puede llevar a cabo mediante un cuchillo filoso, o una pequeña sierra.

La interfase paralelo puede ser armada en algún tipo de plaqueta de armado sin soldadura. Esto nos permitirá solucionar cualquier inconveniente en forma rápida y no corremos el riesgo de arruinar un componente al soldarlo. Recomendamos poner capacitores de 0,1 microfaradio físicamente cerca de cada circuito integrado, entre las líneas de +V y masa, con el objeto de suprimir los picos que puedan aparecer en la fuente, así como las oscilaciones de alta frecuencia.

Una caja adecuada para la interfase puede ser comprada en cualquier comercio de electrónica. Debe ser lo suficientemente grande como para poder contener el circuito y la fuente de alimentación necesaria para la operación de esta interfase.

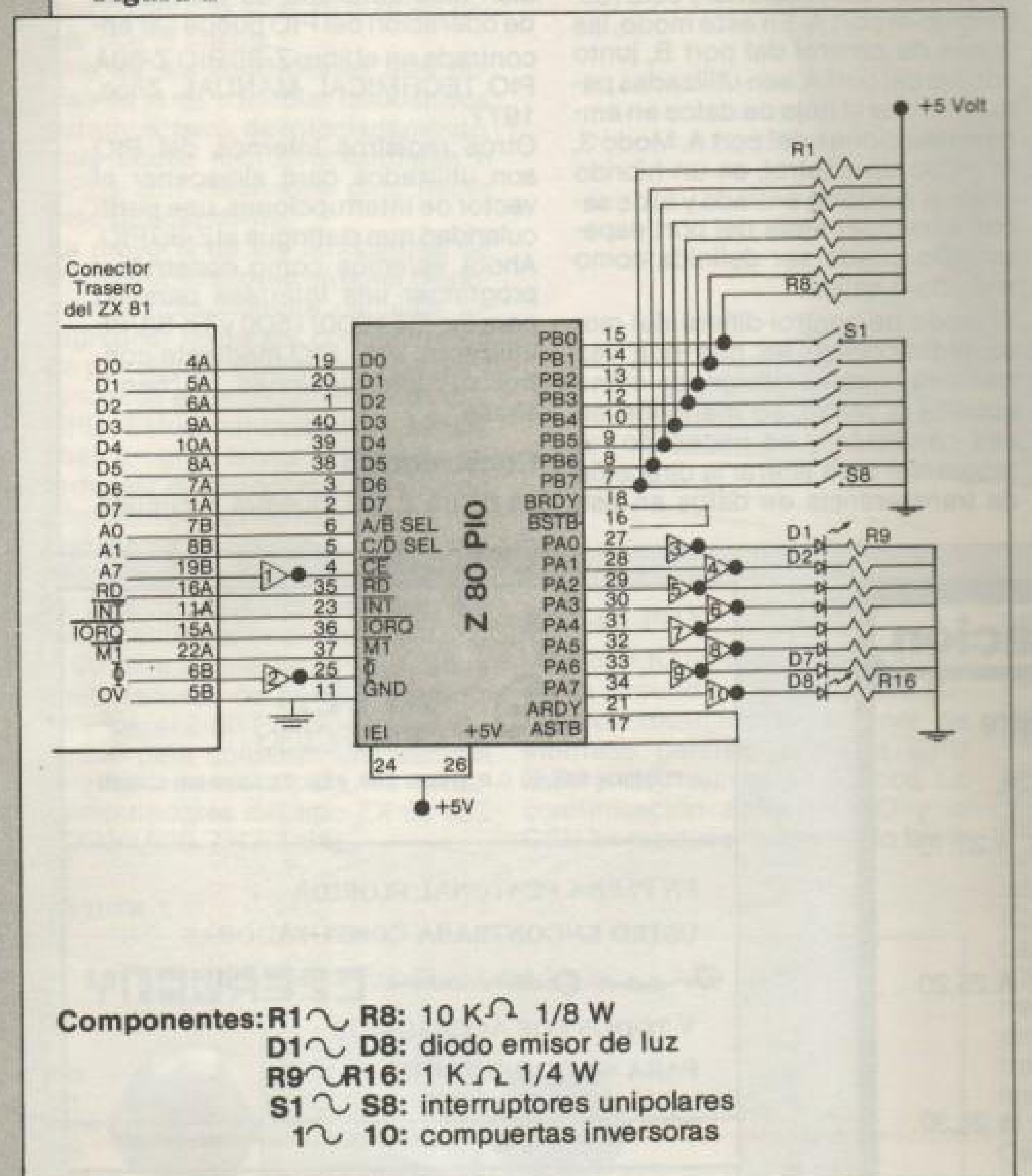
En cuanto a la fuente de alimentación, existen dos opciones.

Si usted planea agregar más circuitos en otro momento, le recomendamos que arme o compre una fuente de alimentación de 5 Volt, 3 Amperes. Un circuito de la misma se puede ver en la figura 3. El regulador de tensión nos asegura que a su salida tendremos siempre 5 Volts, fundamental para no dañar ningún circuito.

Por otra parte, si no planeamos agregar más circuitos al existente, todo lo que necesita es el regulador de 5 Volts, y tomar la tensión necesaria de la alimentación de la máquina. De este modo no podremos exceder un consumo de 300 miliamperes. K64

Continuará

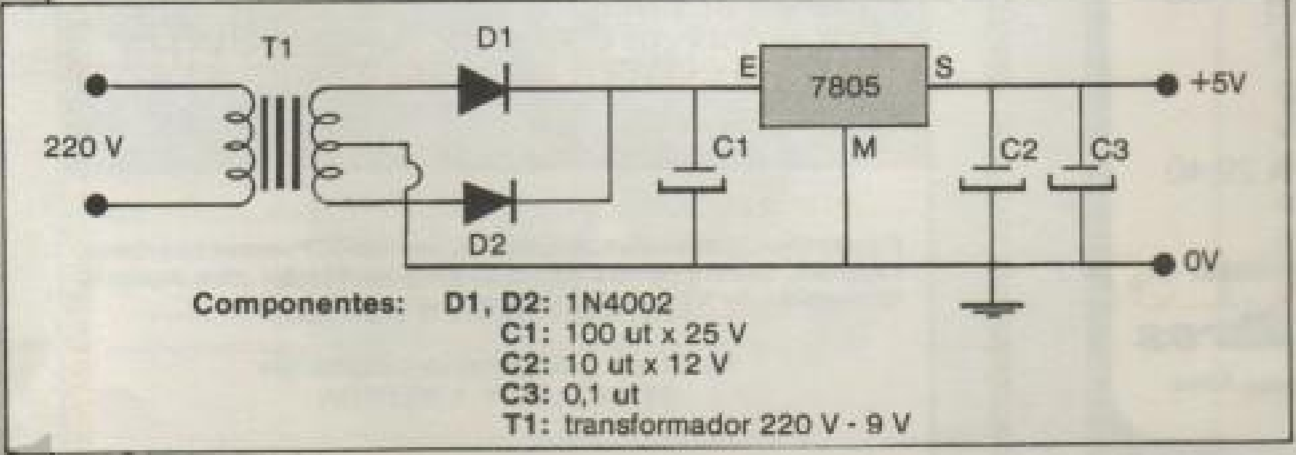
Figura 2





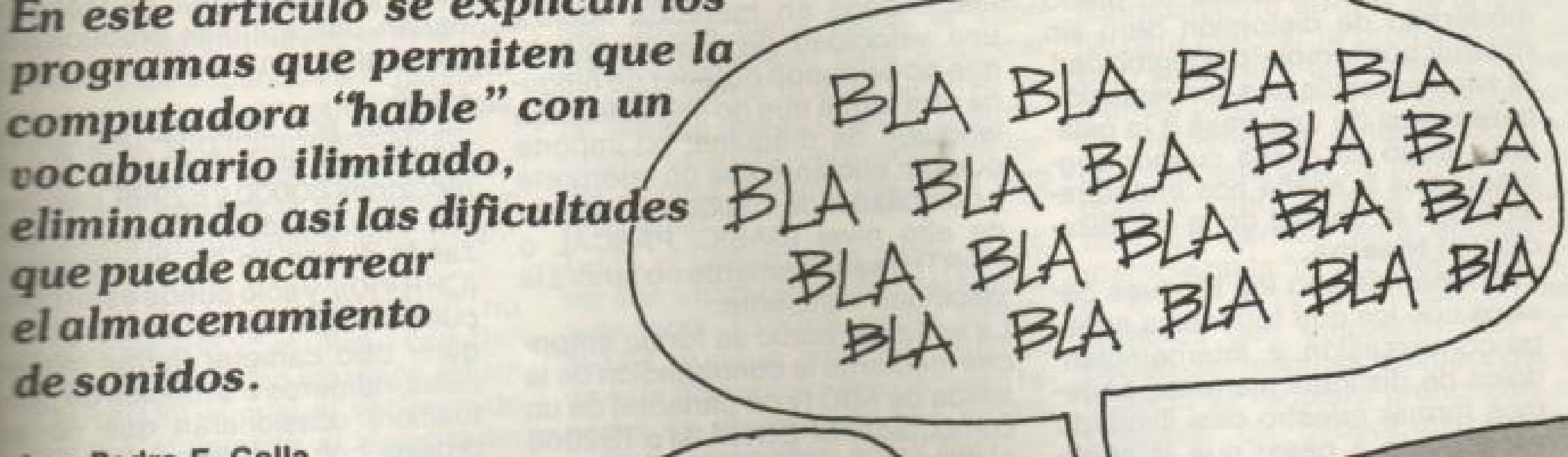
Componentes: R1 ~ R8: 10 KΩ 1/8 W
D1 ~ D8: diodo emisor de luz
R9 ~ R16: 1 KΩ 1/4 W
S1 ~ S8: interruptores unipolares
1 ~ 10: compuertas inversoras

Figura 3



Componentes: D1, D2: 1N4002
C1: 100 ut x 25 V
C2: 10 ut x 12 V
C3: 0,1 ut
T1: transformador 220 V - 9 V

SINTETIZADOR DE VOZ



En este artículo se explican los programas que permiten que la computadora "hable" con un vocabulario ilimitado, eliminando así las dificultades que puede acarrear el almacenamiento de sonidos.

Ing. Pedro E. Colla



Un anterior artículo dedicado al almacenamiento de sonidos, incluyendo voz humana, permitía guardar en memoria segmentos de conversación creando una herramienta mediante la cual se puede hacer "hablar" al computador.

Sin embargo, el mismo no califica como un verdadero sintetizador de voz pues el enganche de los segmentos de habla para formar palabras de un vocabulario más o menos grande es por cierto una tarea ardua.

En este artículo se proporciona un sintetizador de voz de vocabulario ilimitado, el cual no tiene los inconvenientes antes mencionados.

Tal como se había indicado en el artículo anterior, la voz humana se puede considerar en esencia formada por dos componentes; una señal de frecuencia más o menos alta y de amplitud constante mez-

clada con otra de mucho menor frecuencia pero de amplitud variable.
Habíamos visto en esa oportunidad que, el sacrificar este último componente, introducía un grado moderado de distorsión pero sin destruir por completo la legibilidad.
Si segmentamos una conversación pasaremos de una frase a la palabra y luego a la sílaba, cuando segmentamos la sílaba nos encontramos con la unidad de la vocalización, el **fonema.**
Los fonemas son los bloques básicos con los que formamos nuestra conversación, e, intercambiándolos de distintas maneras, podemos formar nuestro casi ilimitado vocabulario a pesar que la cantidad de fonemas son relativamente muy pocos.
En efecto, basta un número algo superior a la cantidad de letras de nuestro abecedario para poder formar sin límite cualquier palabra.
La correspondencia entre letras del alfabeto y fonemas no es uno a uno pues hay letras que, de acuerdo al contexto en que se emplean, requieren la utilización de un fonema distinto; basta, a modo de ejemplo, considerar la utilización de la letra "C" en la palabra **casa** y en la palabra **cena** para comprender este particular.
Si aislamos cada fonema y observamos la forma de onda que tiene la señal de audio se verá que está formada por combinaciones de señales y silencios que se repiten incesantemente.
Un sintetizador de voz que utiliza fonemas recurre a la representación de esas formas características de señal y a su unión en rápida secuencia para formar palabras.
Debido al relativamente reducido número requerido es práctico el almacenamiento de éstos en la memoria de un computador.
El hecho de requerir formar una señal de audio partiendo de datos almacenados en memoria implica una velocidad de procesamiento que no se puede obtener de ninguna otra forma que no sea utilizando lenguaje de máquina; no importa que tan eficiente sea un intérprete o compilador de cualquier lenguaje de alto nivel (BASIC, PASCAL o FORTH) sencillamente no tendrá la velocidad suficiente.
La señal de audio se forma entonces mediante la conmutación de la salida de MIC (y de parlante) de un computador SPECTRUM o TS2068 al ritmo impuesto por datos correspondientes a cada fonema bajo la supervisión de una rutina en lenguaje de máquina.
El programa se puede apreciar en la figura 1; en el mismo la sección en lenguaje de máquina se almacena en instrucciones DATA para facilitar su carga a aquellos que no posean un programa ensamblador.
El programa BASIC almacena el código de máquina en grupos de diez bytes, colocando al comienzo de los mismos la dirección donde deberá cargar el primero de ellos y, al final, la suma de los mismos, de esta manera se facilita notablemente la depuración de eventuales errores de tipeo, comprensibles dado la exagerada extensión.
El programa sintetizador en sí mismo fue extraído de una publicación extranjera y sometido a pequeñas modificaciones francamente cosméticas, por parte del autor.
El mismo se aloja en la parte superior de la memoria RAM ocupando 2200 bytes a partir de la dirección 63000.
Esta ubicación no es casual, la colocación en otra zona inferior implicaría que las interrupciones que recibe el procesador principal por parte del generador de video se harían perceptibles produciendo gangosidad en la salida del sintetizador.
El texto por sintetizar se coloca (pokea) en un buffer entre las direcciones 60000 y 62999, casi media hora de conversación, finalizando él con un "0" hexadecimal (CHR#(0)); y sólo puede estar compuesto de letras mayúsculas y cualquier otro caracter (letras minúsculas, números o símbolos de puntuación) ocasionarán que no se prosiga con el remanente a partir de allí.
Esto representa cierta incomodidad en el uso práctico pues al margen de la limitación en cuanto al tipo de caracteres se requiere agregar una pequeña sección de programa BASIC que se encargue de armar el buffer para cada uso del sintetizador.
La solución viene de la mano de una corta rutina en lenguaje de máquina el cual funciona como "driver" para integrar el programa en la arquitectura del computador y permitir acceder al mismo a partir de la instrucción PRINT.
Para lograr esto se modifica el área de variable de sistemas denominada CHANS reemplazando las direcciones de manejo de dispositivos tipo "P" (Printer) por la dirección de comienzo del driver. Cada vez que se emplee LPRINT en vez de recurrir a la rutina en ROM destinada al efecto el caracter será enviado al driver.
Esta técnica fue profusamente explicada y empleada en un artículo anterior para el manejo de una im-



presora Centronics.

El driver se encarga de colocar el caracter en el buffer manejando la utilización de letras minúsculas y caracteres especiales; los primeros son transformados en mayúsculas y los segundos son reemplazados por espacios pues, de todas maneras, son impronunciables.

Los números son reemplazados por una secuencia de caracteres prevista en una tabla al efecto dentro del driver.

Cuando la sentencia LPRINT no concluye con ";" el último caracter enviado es un (cr) (carriage return CHR#(13)) el cual al ser recibido implica el llamado al sintetizador en sí mismo.

Si se utiliza ";" el programa asume que seguirán viniendo caracteres y maneja la situación "recordando" la próxima posición que tiene que utilizar en el buffer.

El driver también está incluído como sentencias DATA en el programa BASIC en la Fig. 1 y para referencia la expresión del mismo en Assembler puede visualizarse en la figura 2; aunque cabe remarcar que este último es incluído solamente en forma indicativa pues bastará con utilizar las sentencias DATA incluidas en la Fig. 1 para generar el segmento de código de máquina respectivo; al igual que el código del sintetizador en caso de haber problemas se exhibirá durante la carga del código el número de línea en que se encuentra la sentencia DATA con problemas.

Cuando logran cargar totalmente sintetizador y driver el mismo programa borra todas las sentencias DATA y las secciones de código destinadas a su carga en memoria quedando una versión reducida que se salva en cassette para uso futuro en la forma de un pequeño programa BASIC y dos segmentos "CODE" conteniendo respectivamente el sintetizador y el driver.

En la sección superviviente de BASIC queda almacenada la información necesaria para modificar en cada carga el área CHANS de tal manera que al finalizar el sintetizador esté listo para su uso en forma aislada o como parte de otro programa.

Tanto el sintetizador como el driver son utilizables tanto para computadores TS2068 como SPECTRUM, aunque para los usuarios de esta última máquina deberán tenerse en cuenta algunas modificaciones menores; estas son:

1) El área CHANS está ubicada en los computadores SPECTRUM ligeramente desplazada con respecto a la TS2068 y por lo tanto se deberán modificar las líneas 120 a 150 para reflejar esto, debiéndose tipear en lugar de lo indicado por la figura 1, lo siguiente:

```
120 POKE 23749,216
130 POKE 23750,214
140 POKE 23751,216
150 POKE 23752,214
```

2) El SPECTRUM no posee la sentencia DELETE para borrar instrucciones de un programa BASIC, por lo que no podrá ejecutar las líneas 525 a 546 de la Fig. 1, para obtener una versión reducida del sintetizador apta en el uso frecuente se deberá borrar "a mano" las líneas indicadas por las instrucciones DELETE y realizar los SAVE de líneas 550 a 570 no como parte del programa sino en forma manual.

Con las pequeñas precauciones derivadas de los puntos anteriores no existe otra dificultad en la utilización de este programa por igual en ambas máquinas.

Por último, nótese que, como parte del programa de carga, se hace decir al sintetizador algunos mensajes.

En los mismos se han utilizado deliberadamente errores de ortografía para mejorar la pronunciación; esto se debe a que solamente se maneja un fonema por cada caracter debiéndose utilizar con frecuencia este tipo de manejo para obtener una correcta pronunciación. En general será necesario reemplazar la "c" y "z" por "s", la "k" o "q" por "c" y la "v" por "b"; las palabras que empiezan con "h" deben ser escritas omitiéndola y cuando se requiera pronunciar el conjunto "qu" deberá utilizarse la "q" únicamente.

A pesar de estas precauciones se obtendrá pronunciación algo deficiente en determinados contextos para las letras "y", "j", "g", "l". Las palabras que utilicen "ch", "gu", "ll" o la letra "y" al comienzo no serán tampoco demasiado naturales.

Pese a las deficiencias apuntadas y al hecho que de todas maneras la pronunciación obtenida es a menudo difícil de interpretar el proyecto es interesante y obviamente de muy bajo costo.

Por otra parte el lector que se aventure al no poco tedioso tipeo necesario notará un efecto de acostumbramiento al cabo de cierto tiempo de uso que hará que el mismo suene cada vez más natural a medida que pasa el tiempo de uso del mismo, incluyendo la detección de cierta "tonada" española que hace inconfundible el origen del programa. K64

Aclaración: las líneas 525 a 546 por un error de impresión omiten el comando "DELETE" el cual debe ser agregado por los lectores.

```
10 REM ----------------------
20 REM SPeech
30 REM Sintetizador de Voz
40 REM ----------------------
50 REM Abre canal #4
60 REM ----------------------
70 OPEN # 4,"P"
80 REM ----------------------
85 REM Genera direccion CHANS
87 REM Para el driver el mismo
88 REM se coloca en 55000
90 REM ----------------------
100 POKE 23582,16
110 POKE 23583,0
120 POKE 26703,216
130 POKE 26704,214
140 POKE 26705,216
150 POKE 26706,214
152 GO TO 160
154 LOAD "" CODE 55000
156 LOAD "" CODE 53000
160 REM ----------------------
170 REM Carga Sintetizador
180 REM ----------------------
185 CLS : CLEAR 54999
190 RESTORE 1000
195 PRINT  AT 0,0; FLASH 1;"Sintetizador -->"; FLASH 0;
200 READ adress
210 IF adress=99999 THEN  GO TO 340
215 LET t=0
220 FOR i=0 TO 14
230 READ byte
240 PRINT  AT 0,25;adress+i
250 POKE adress+i,byte
260 LET t=t+byte
270 NEXT i
280 READ sum
290 IF t <> sum THEN  PRINT "ERROR EN LINEA "
   adress-62000: STOP
300 GO TO 200
310 REM ----------------------
320 REM Driver del Canal
330 REM ----------------------
340 RESTORE 5000
345 PRINT  AT 1,0; FLASH 1;"Sintetizador -->"; FLASH 0;
350 READ adress
360 IF adress=99999 THEN  GO TO 500
365 PRINT  AT 1,25;adress+i
370 LET t=0
380 FOR i=0 TO 14
390 READ byte
400 POKE adress+i,byte
410 LET t=t+byte
420 NEXT i
430 READ sum
440 IF t <> sum THEN  PRINT "Error en Linea "
   adress-50000: STOP
450 GO TO 350
500 PRINT #4;"a finalisado la karga 0123456789"
520 PRINT  AT 3,0; FLASH 1;"Colocar un Cassette"; FLASH 0;
522 GO TO 530
524 PAPER 7: INK 0: BORDER 0: CLS : PRINT  AT 0,0; FLASH
   1;"Sintetizador Listo"
 FLASH 0: PRINT #4;"sintetisador listo"
525 10,525
530 160,180
535 1000,9000
540 190,530
545 152,152
546 522,522
550 SAVE "SPeech" LINE 10
560 SAVE "SPeech.DRV" CODE 55000,250
570 SAVE "SPeech.RUT" CODE 63000,2200
```

```
4980 REM ----------------------
4990 REM CODIGO DEL DRIVER
4995 REM ----------------------
5000 DATA 55000,195,222,214,000,116,234,245,058,219,214,1717
5010 DATA 55010,254,255,032,006,042,220,214,000,024,003,1050
5020 DATA 55020,033,096,234,241,245,254,013,040,034,214,1404
5030 DATA 55030,048,056,040,241,245,214,058,056,049,241,1248
5040 DATA 55040,245,214,065,056,028,241,245,214,091,056,1455
5050 DATA 55050,029,241,245,214,097,056,016,214,123,056,1291
5060 DATA 55060,022,024,010,241,175,050,219,214,119,205,1279
5070 DATA 55070,024,246,201,241,062,032,119,035,024,042,1026
5080 DATA 55080,241,024,249,241,214,032,024,244,241,071,1581
5090 DATA 55090,229,033,093,215,126,184,040,011,126,254,1311
5100 DATA 55100,255,040,003,035,024,248,035,024,241,035,0940
5110 DATA 55110,235,225,026,254,255,040,005,119,019,035,1213
5120 DATA 55120,024,246,034,220,214,035,175,119,061,050,1178
5130 DATA 55130,219,214,201,049,085,078,079,032,255,050,1262
5140 DATA 55140,068,079,083,032,255,051,084,082,069,083,0886
5150 DATA 55150,032,255,052,075,085,065,084,082,079,032,0841
5160 DATA 55160,255,053,083,073,078,075,079,032,255,054,1037
5170 DATA 55170,083,069,073,083,032,255,055,083,073,069,0875
5180 DATA 55180,084,069,032,255,056,079,067,072,079,032,0825
5190 DATA 55190,255,057,078,085,069,086,069,032,255,048,1034
5200 DATA 55200,083,069,082,079,032,255,255,255,000,000,1110
5210 DATA 55210,000,000,000,000,000,000,000,000,000,000,0000
5220 DATA 99999
```

```
990 REM ----------------------
995 REM CODIGO DEL SINTETIZADOR
997 REM ----------------------
1000 DATA 63000,195,177,246,243,078,035,197,229,094,035,1521
1010 DATA 63010,086,035,070,120,167,040,009,062,020,062,0674
1020 DATA 63020,254,205,176,246,016,247,035,078,062,007,1318
1030 DATA 63030,063,254,205,176,246,016,247,035,027,122,1391
1040 DATA 63040,179,032,225,225,193,016,218,251,201,007,1844
1050 DATA 63050,050,000,000,005,006,024,004,002,001,003,0095
1060 DATA 63060,002,025,009,007,002,002,009,022,006,010,0114
1070 DATA 63070,004,000,004,041,006,040,000,026,004,020,0166
1080 DATA 63080,001,001,001,006,001,019,001,005,007,026,0060
1090 DATA 63090,002,004,002,010,001,003,001,029,004,001,0087
1100 DATA 63100,006,019,002,002,002,006,003,026,007,034,0109
1110 DATA 63110,010,004,001,010,002,006,003,016,002,010,0078
1120 DATA 63120,003,012,001,011,005,020,002,011,010,004,0082
1130 DATA 63130,001,010,002,002,001,006,007,010,002,010,0051
1140 DATA 63140,002,004,001,003,002,006,005,016,002,008,0051
1150 DATA 63150,003,010,201,002,167,245,033,096,234,241,1300
1160 DATA 63160,015,050,043,246,050,054,246,126,254,032,1116
1170 DATA 63170,040,052,254,065,218,165,247,254,123,210,1640
1180 DATA 63180,165,247,254,091,056,007,254,097,218,165,1594
1190 DATA 63190,247,214,032,197,229,214,065,203,039,095,1535
1200 DATA 63200,022,000,033,004,247,025,094,035,102,107,0669
1210 DATA 63210,205,027,246,225,193,011,035,120,177,202,1441
1220 DATA 63220,165,247,024,199,197,229,033,064,031,043,1252
1230 DATA 63230,124,181,032,251,024,233,056,247,055,248,1451
1240 DATA 63240,130,250,049,250,195,247,008,253,124,254,1848
1250 DATA 63250,096,201,220,247,073,246,088,249,013,254,1737
1260 DATA 63260,037,249,146,248,073,247,244,247,088,249,1828
1270 DATA 63270,181,248,202,247,211,252,237,247,169,249,2043
1280 DATA 63280,092,248,096,251,079,251,130,250,013,007,1425
1290 DATA 63290,002,020,024,023,035,007,078,020,044,029,0298
1300 DATA 63300,050,023,045,018,015,014,003,000,023,096,0297
1310 DATA 63310,029,131,027,162,010,050,000,003,007,004,0423
1320 DATA 63320,009,002,009,002,007,002,000,001,012,002,0064
1330 DATA 63330,013,004,007,005,009,004,017,000,007,004,0070
1340 DATA 63340,009,003,012,001,023,003,007,005,006,005,0074
1350 DATA 63350,005,002,010,000,009,003,023,001,006,004,0063
1360 DATA 63360,012,001,007,004,021,003,031,004,008,005,0096
1370 DATA 63370,010,003,013,002,010,000,004,001,009,004,0055
1380 DATA 63380,009,003,000,004,032,002,009,004,007,004,0082
1390 DATA 63390,007,004,005,004,005,000,014,003,006,007,0061
1400 DATA 63400,008,004,009,004,012,004,010,003,007,003,0064
1410 DATA 63410,019,003,025,003,008,001,032,062,062,050,0265
1420 DATA 63420,043,246,060,050,054,246,201,013,011,000,0924
1430 DATA 63432,005,093,013,006,023,110,007,019,006,019,0301
1440 DATA 63440,010,013,022,014,000,010,007,022,000,011,0134
```

```
1450 DATA 63450,022,012,013,007,000,010,193,019,003,020,0299
1460 DATA 63460,001,003,013,006,100,010,006,016,009,010,0262
1470 DATA 63470,002,000,027,149,033,127,001,032,000,000,0371
1480 DATA 63480,255,000,255,000,255,000,255,000,255,000,1275
1490 DATA 63490,255,000,255,000,255,000,255,000,255,000,1275
1500 DATA 63500,255,000,255,000,255,000,255,000,255,000,1275
1510 DATA 63510,255,000,255,000,255,000,255,000,255,000,1275
1520 DATA 63520,161,024,255,000,229,000,019,036,064,000,0708
1530 DATA 63530,073,036,153,017,010,006,004,021,015,005,0420
1540 DATA 63540,106,023,041,001,017,000,000,255,000,255,0698
1550 DATA 63550,000,255,000,255,000,255,000,161,024,255,1205
1560 DATA 63560,000,229,000,019,036,064,000,073,036,153,0610
1570 DATA 63570,017,010,006,004,021,015,005,106,023,041,0328
1580 DATA 63580,009,003,000,028,027,025,140,044,185,001,0470
1590 DATA 63590,021,000,001,060,034,224,010,010,002,002,0360
1600 DATA 63600,001,004,007,059,000,179,032,001,040,094,0505
1610 DATA 63610,007,007,031,026,000,033,006,175,032,000,0413
1620 DATA 63620,047,106,029,226,029,096,000,255,000,255,1043
1630 DATA 63630,000,255,000,255,001,072,000,009,005,013,0611
1640 DATA 63640,010,016,032,000,128,006,034,009,032,012,0295
1650 DATA 63650,130,013,030,014,066,009,034,009,038,009,0352
1660 DATA 63660,016,009,020,010,038,012,132,004,034,009,0284
1670 DATA 63670,032,013,026,002,000,000,044,006,064,009,0292
1680 DATA 63680,032,013,034,000,014,009,006,013,010,016,0163
1690 DATA 63690,032,000,128,006,034,009,032,012,130,013,0404
1700 DATA 63700,030,014,066,009,034,009,030,009,016,009,0234
1710 DATA 63710,020,010,030,012,132,004,034,009,032,013,0304
1720 DATA 63720,026,002,000,006,044,006,064,009,032,013,0292
1730 DATA 63730,034,000,014,009,006,013,010,016,032,000,0158
1740 DATA 63740,128,006,034,009,032,012,130,013,030,014,0408
1750 DATA 63750,066,009,034,009,038,009,016,009,020,010,0228
1760 DATA 63760,030,012,132,004,034,009,032,013,026,002,0294
1770 DATA 63770,000,000,044,006,064,009,032,013,034,000,0206
1780 DATA 63780,014,006,024,000,009,006,013,010,016,032,0130
1790 DATA 63790,000,128,006,034,009,032,012,130,013,030,0402
1800 DATA 63800,014,066,009,034,009,030,009,016,009,020,0224
1810 DATA 63810,010,030,012,132,004,034,009,032,013,026,0310
1820 DATA 63820,002,000,000,044,006,064,009,032,013,034,0300
1830 DATA 63830,000,014,001,044,000,000,255,000,255,000,0577
1840 DATA 63840,255,000,255,000,255,000,255,000,255,000,1275
1850 DATA 63850,255,000,255,000,255,000,255,000,255,000,1275
1860 DATA 63860,255,000,255,000,255,000,255,000,255,000,1275
1870 DATA 63870,255,000,255,000,255,001,065,020,040,013,0904
1880 DATA 63880,042,012,005,001,033,017,046,006,047,007,0216
1890 DATA 63890,045,011,233,007,006,006,030,022,041,014,0415
1900 DATA 63900,035,024,031,015,029,025,037,027,027,021,0271
1910 DATA 63910,032,225,201,001,065,000,000,100,000,255,0959
1920 DATA 63920,000,203,001,033,000,156,000,029,000,053,0475
1930 DATA 63930,001,077,000,047,000,056,001,072,000,027,0281
1940 DATA 63940,000,164,001,020,000,019,002,052,002,015,0275
1950 DATA 63950,001,026,000,007,000,030,002,072,001,046,0193
1960 DATA 63960,000,013,002,024,000,021,000,028,003,032,0123
1970 DATA 63970,002,000,003,016,003,004,002,007,003,010,0146
1980 DATA 63980,000,005,003,017,002,026,001,061,003,022,0140
1990 DATA 63990,000,006,003,015,003,022,000,005,000,006,0060
2000 DATA 64000,000,053,000,030,002,019,001,014,001,006,0134
2010 DATA 64010,002,075,002,080,001,021,000,069,000,066,0316
2020 DATA 64020,002,004,002,073,007,006,001,041,002,234,0372
2030 DATA 64030,014,051,005,001,003,007,002,005,005,006,0099
2040 DATA 64040,004,004,001,166,013,204,062,003,014,001,0472
2050 DATA 64050,043,000,000,097,006,106,006,029,009,031,0327
2060 DATA 64060,010,015,002,010,002,017,006,017,002,042,0131
2070 DATA 64070,000,016,008,013,007,014,005,016,006,155,0240
2080 DATA 64080,000,030,007,036,002,016,008,017,008,010,0142
2090 DATA 64090,010,024,010,052,001,000,000,066,008,014,0185
2100 DATA 64100,032,010,003,015,000,093,008,001,000,002,0172
2110 DATA 64110,014,006,005,105,002,000,040,126,042,172,0600
2120 DATA 64120,036,182,039,085,007,065,001,028,002,004,0449
2130 DATA 64130,005,004,000,112,040,151,035,146,002,097,0600
2140 DATA 64140,000,000,038,002,022,002,043,001,056,001,0165
2150 DATA 64150,039,001,022,001,003,005,070,002,003,002,0148
2160 DATA 64160,012,001,021,001,005,001,027,001,062,002,0133
2170 DATA 64170,037,009,010,001,004,001,004,002,011,001,0080
2180 DATA 64180,017,001,005,001,065,001,004,007,051,001,0153
2190 DATA 64190,052,001,006,001,020,001,006,001,020,001,0109
2200 DATA 64200,002,001,005,001,010,001,021,001,064,001,0195
2210 DATA 64210,036,001,031,001,196,001,122,001,016,001,0406
2220 DATA 64220,056,001,006,001,136,001,021,002,004,001,0229
2230 DATA 64230,101,001,031,001,023,001,040,001,006,001,0206
2240 DATA 64240,036,001,007,001,006,003,056,001,004,002,0117
2250 DATA 64250,042,001,011,001,014,001,004,001,051,001,0127
2260 DATA 64260,028,002,096,002,003,001,014,002,006,001,0155
2270 DATA 64270,085,001,010,004,001,002,013,001,021,002,0140
2280 DATA 64280,252,003,020,002,003,003,094,002,040,001,0420
2290 DATA 64290,063,001,007,001,004,001,012,001,080,001,0171
2300 DATA 64300,005,001,013,001,004,001,040,001,014,001,0081
2310 DATA 64310,019,004,004,001,009,002,001,004,020,001,0065
2320 DATA 64320,062,001,019,001,006,002,072,001,110,005,0279
2330 DATA 64330,036,001,020,001,028,009,007,000,010,193,0305
2340 DATA 64340,019,003,020,001,003,013,006,100,010,006,0261
2350 DATA 64350,016,009,001,186,000,000,255,000,255,000,0722
2360 DATA 64360,255,000,255,000,255,000,255,000,255,000,1275
2370 DATA 64370,255,000,255,000,255,000,255,000,255,000,1275
2380 DATA 64380,255,000,255,000,255,000,255,000,255,000,1275
2390 DATA 64390,255,000,255,000,255,000,255,000,255,000,1275
2400 DATA 64400,255,000,255,000,255,001,065,020,040,013,0904
2410 DATA 64410,042,012,005,001,033,017,046,006,047,007,0216
2420 DATA 64420,045,011,233,007,006,006,030,022,041,014,0415
2430 DATA 64430,035,024,031,015,029,025,037,027,027,021,0271
2440 DATA 64440,032,003,007,004,009,002,009,002,007,002,0077
2450 DATA 64450,000,001,012,002,013,004,007,005,009,004,0065
2460 DATA 64460,017,000,007,004,009,003,012,001,023,003,0079
2470 DATA 64470,007,005,006,005,005,002,010,000,009,003,0052
2480 DATA 64480,023,001,006,004,012,001,007,004,021,003,0082
2490 DATA 64490,031,004,008,005,010,003,013,002,010,000,0086
2500 DATA 64500,004,001,008,004,009,003,008,004,032,002,0075
2510 DATA 64510,009,004,007,004,007,004,008,004,008,000,0055
2520 DATA 64520,014,003,006,007,008,004,009,004,012,004,0071
2530 DATA 64530,010,003,007,003,019,003,007,004,009,002,0067
2540 DATA 64540,009,002,007,002,008,001,012,002,013,004,0060
2550 DATA 64550,007,005,009,004,017,000,007,004,009,003,0065
2560 DATA 64560,012,001,023,003,007,005,006,005,005,002,0069
2570 DATA 64570,010,000,009,003,023,001,006,004,012,001,0069
2580 DATA 64580,007,004,021,003,031,004,008,005,010,003,0096
2590 DATA 64590,013,002,010,000,004,001,008,004,009,003,0054
2600 DATA 64600,008,004,032,002,009,004,007,004,007,004,0081
2610 DATA 64610,008,004,008,000,014,003,006,007,008,004,0062
2620 DATA 64620,009,004,012,004,010,003,007,003,019,003,0074
2630 DATA 64630,007,004,009,002,009,002,007,002,008,001,0051
2640 DATA 64640,012,002,013,004,007,005,009,004,017,000,0073
2650 DATA 64650,007,004,009,003,012,001,023,003,007,005,0074
2660 DATA 64660,006,005,005,002,010,000,009,003,023,001,0064
2670 DATA 64670,006,004,012,001,007,004,021,003,031,004,0093
2680 DATA 64680,008,005,010,003,013,002,010,000,004,001,0056
2690 DATA 64690,008,004,009,003,008,004,032,002,009,004,0083
2700 DATA 64700,007,004,007,004,008,004,008,000,014,003,0059
2710 DATA 64710,006,007,008,004,009,004,012,004,010,003,0067
2720 DATA 64720,007,003,019,001,065,000,000,255,000,255,0605
2730 DATA 64730,000,255,000,255,000,255,000,255,000,255,1275
2740 DATA 64740,000,255,000,255,000,255,000,255,000,255,1275
2750 DATA 64750,000,255,000,255,000,255,000,255,000,255,1275
2760 DATA 64760,000,255,000,255,000,255,000,232,008,024,1029
2770 DATA 64770,011,015,006,015,009,016,010,016,007,016,0121
2780 DATA 64780,010,016,007,130,005,021,008,019,009,139,0372
2790 DATA 64790,004,067,015,012,008,015,009,013,005,034,0182
2800 DATA 64800,010,014,021,030,009,001,004,026,010,033,0158
2810 DATA 64810,007,055,004,037,006,167,032,023,010,025,0366
2820 DATA 64820,007,004,012,149,054,001,001,000,000,002,0310
2830 DATA 64830,024,236,008,007,013,008,029,013,017,007,0442
2840 DATA 64840,007,128,001,004,008,149,053,170,010,004,0542
2850 DATA 64850,004,123,034,023,005,161,001,090,000,000,0441
2860 DATA 64860,180,000,255,000,203,001,033,001,156,001,0830
2870 DATA 64870,029,001,053,001,077,001,047,001,072,001,0283
2880 DATA 64880,027,001,164,001,020,001,019,002,052,002,0289
2890 DATA 64890,015,001,026,001,007,001,030,002,072,001,0164
2900 DATA 64900,046,001,013,002,024,001,021,001,028,003,0140
2910 DATA 64910,032,002,008,003,016,003,004,002,007,003,0160
2920 DATA 64920,018,001,005,003,017,002,026,001,061,003,0137
2930 DATA 64930,022,000,006,003,015,003,022,001,005,001,0078
2940 DATA 64940,006,001,053,001,030,002,019,001,014,001,0136
2950 DATA 64950,006,002,075,002,080,001,021,001,069,001,0258
2960 DATA 64960,066,002,004,002,073,007,006,001,041,002,0204
2970 DATA 64970,234,014,051,005,001,003,007,002,005,005,0327
2980 DATA 64980,006,004,001,166,013,204,062,255,000,0715
2990 DATA 64990,255,000,103,029,179,012,006,016,002,006,0608
3000 DATA 65000,002,013,078,039,156,019,003,015,004,015,0344
3010 DATA 65010,003,032,159,021,001,016,005,010,092,033,0452
3020 DATA 65020,163,019,004,013,007,004,002,006,094,033,0345
3030 DATA 65030,166,013,005,013,006,014,094,001,054,000,0366
3040 DATA 65040,005,001,002,027,014,003,001,036,003,255,0347
3050 DATA 65050,000,002,020,024,010,002,008,029,017,040,0312
3060 DATA 65060,000,255,000,057,022,024,005,000,003,002,0448
3070 DATA 65070,012,003,001,021,019,042,002,255,000,064,0419
3080 DATA 65080,021,027,007,090,016,026,022,035,001,255,0500
3090 DATA 65090,000,044,021,025,001,097,021,024,014,255,0502
3100 DATA 65100,000,094,022,137,022,027,014,255,000,080,0651
3110 DATA 65110,022,033,005,047,014,029,027,024,018,037,0256
3120 DATA 65120,001,004,001,037,015,037,000,008,001,037,0141
3130 DATA 65130,002,003,002,034,012,010,001,014,001,005,0084
3140 DATA 65140,001,005,000,011,001,004,002,003,001,014,0042
3150 DATA 65150,000,000,255,000,255,000,255,000,255,000,1020
3160 DATA 65160,255,000,255,000,255,000,255,010,200,100,1330
3170 DATA 65170,100,010,200,100,100,010,200,100,100,000,0920
3180 DATA 65180,000,000,000,000,000,000,000,000,000,000,0000
3190 DATA 65190,000,000,000,000,000,000,000,000,000,000,0000
3200 DATA 65200,000,000,000,000,000,000,000,000,000,000,0000
4000 DATA 99999
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

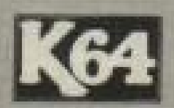
```
00010         ORG 55000
00020 ;-----------------------
00030 ;Speech.DRV
00040 ;
00050 ;DRIVER DEL CANAL #4 PARA
00060 ;SINTETIZADOR DE VOZ
00070 ;
00080 ;-----------------------
00090 BEGIN  JP MAIN
00100 FLAG   DEFB #00
00110 ADRESS DEFB #00,#00
00120 ;-----------------------
00130 ;CUERPO PRINCIPAL
00140 ;-----------------------
00150 MAIN   PUSH AF
00160 ;-----------------------
00170 ;VERIFICA SI ES EL PRIMERO
00180 ;-----------------------
00190        LD A,(FLAG)
00200        CP #FF
00210        JR NZ,FIRST
00220 FOLLOW LD HL,(ADRESS)
00230        NOP
00240        JR PUT
00250 FIRST  LD HL,60000
00260 ;-----------------------
00270 ;CARGA EN EL BUFFER
00280 ;-----------------------
00290 PUT    POP AF
00300        PUSH AF
00310 ;-----------------------
00320 ;VERIFICA SI NO ES cr
00330 ;EN CUYO CASO LLAMA AL
00340 ;SINTETIZADOR Y ACOMODA
```



```
00350 ;flag PARA EL PROXIMO USO
00360 ;COLOQUE SU PRIMER BYTE
00370 ;EN 60000
00380 ;-----------------------
00390         CP 13
00400         JR Z,CR
00410 ;-----------------------
00420 ;EN CASO DE NO SER cr
00430 ;DESVIA A DISTINTAS
00440 ;RUTINAS DE ACUERDO AL
00450 ;CARACTER QUE SE TRATE
00460 ;ESTOS PUEDEN SER
00470 ;
00480 ; MAYUSCULAS
00490 ; MINUSCULAS
00500 ; NUMEROS
00510 ;
00520 ;LOS CARACTERES DE
00530 ;CONTROL Y ESPECIALES SON
00540 ;REEMPLAZADOS POR SPACES
00550 ;LOS NUMEROS SON
00560 ;CONVERTIDOS MEDIANTE UNA
00570 ;TABLA
00580 ;-----------------------
00590         SUB 48
00600         JR C,SPACE
00610         POP AF
00620         PUSH AF
00630         SUB 58
00640         JR C,NUMS
00650         POP AF
00660         PUSH AF
00670         SUB 65
00680         JR C,SPACE
00690         POP AF
00700         PUSH AF
00710         SUB 91
00720         JR C,LETS
00730         POP AF
00740         PUSH AF
00750         SUB 97
00760         JR C,SPACE
00770         SUB 123
00780         JR C,MINS
00790         JR SPACE
00800 ;-----------------------
00810 ;RUTINA DE cr
00820 ;-----------------------
00830 CR      POP AF
00840         XOR A
00850         LD (FLAG),A
00860         LD (HL),A
00870         CALL 63000
00880         RET
00890 ;-----------------------
00900 ;RUTINA DE ESPACIOS
00910 ;-----------------------
00920 SPACE   POP AF
00930         LD A,"
00940 SPAC1   LD (HL),A
00950         INC HL
00960         JR FIN
00970 ;-----------------------
00980 ;RUTINA DE MAYUSCULAS
00990 ;-----------------------
01000 LETS    POP AF
01010         JR SPAC1
01020 ;-----------------------
01030 ;RUTINA DE MINUSCULAS
01040 ;-----------------------
01050 MINS    POP AF
01060         SUB 32
01070         JR SPAC1
01080 ;-----------------------
01090 ;RUTINA DE NUMEROS
01100 ;-----------------------
01110 NUMS    POP AF
01120         LD B,A
01130         PUSH HL
01140         LD HL,TABLA
01150 NUM1    LD A,(HL)
01160         CP B
01170         JR Z,NUMZ
01180 NUM2    LD A,(HL)
01190         CP #FF
01200         JR Z,FIND
01210         INC HL
01220         JR NUM2
01230 FIND    INC HL
01240         JR NUM1
01250 NUMZ    INC HL
01260         EX DE,HL
01270         POP HL
01280 NUMZ1   LD A,(DE)
01290         CP #FF
01300         JR Z,FIN
01310         LD (HL),A
01320         INC DE
01330         INC HL
01340         JR NUMZ1
01350 ;-----------------------
01360 ;RUTINA DE FINALIZACION
01370 ;ESTA ES INVOCADA SOLO
01380 ;SI LA INSTRUCCION DE
01390 ;IMPRESION FINALIZA CON
01400 ; ";" EN CUYO CASO EL
01410 ;DRIVER ASUME QUE
01420 ;CONTINUARAN LOS
01430 ;CARACTERES
01440 ;-----------------------
01450 FIN     LD (ADRESS),HL
01460         INC HL
01470         XOR A
01480         LD (HL),A
01490         DEC A
01500         LD (FLAG),A
01510         RET
01520 ;-----------------------
01530 ;TABLA DE CONVERSION DE
01540 ;NUMEROS
01550 ;-----------------------
01560 TABLA   DEFB "1
01570         DEFM /UNO /
01580         DEFB #FF,"2
01590         DEFM /DOS /
01600         DEFB #FF,"3
01610         DEFM /TRES /
01620         DEFB #FF,"4
01630         DEFM /KUATRO /
01640         DEFB #FF,"5
01650         DEFM /SINKO /
01660         DEFB #FF,"6
01670         DEFM /SEIS /
01680         DEFB #FF,"7
01690         DEFM /SIETE /
01700         DEFB #FF,"8
01710         DEFM /OCHO /
01720         DEFB #FF,"9
01730         DEFM /NUEVE /
01740         DEFB #FF,"0
01750         DEFM /SERO /
01760         DEFB #FF,#FF,#FF
01770 FINAL   NOP

S
FINAL      = #D7A8
NUMZ1      = #D748
FIND       = #D742
NUM2       = #D73A
NUMZ       = #D745
NUM1       = #D736
TABLA      = #D75D
FIN        = #D752
SPAC1      = #D724
MINS       = #D72B
LETS       = #D728
NUMS       = #D730
SPACE      = #D721
CR         = #D717
PUT        = #D6EF
FOLLOW     = #D6E6
FIRST      = #D6EC
ADRESS     = #D6DC
FLAG       = #D6DB
MAIN       = #D6DE
BEGIN      = #D6D8

T
Start of source = 32768
Length          = 02753
```

GUIA PRACTICA

20 JUEGOS PARA TK 90X 48K
TS 2068 - SPECTRUM = ₳ 5
JOYSTICKS + 4 JUEGOS = ₳ 20

COMPRA - VENTA - CANJE: COMPUTADORAS - T.V. - VIDEOCASSETERAS

COMPUPAL — SERRANO 1980 CAP.

COMMODORE 64-128

PROGRAMAS DE APLICACION. STOCK. ARCHIVOS. LISTAS. CUENTAS CORRIENTES. VENTAS. BASES DE DATOS. LO ULTIMO EN JUEGOS. MAS DE 1500 TITULOS. JOYSTICKS. DATASETTE. MANUAL EN CASTELLANO P/128. SERVICIO TECNICO ESPECIALIZADO.

CHIPS COMPUTACION

RODRIGUEZ PEÑA 770, 9º "49" TE: 42-3589

COMMODORE 64 - 128

- 1600 Titulos en Juegos - 300 Utilitarios
- Libros: 70 Titulos en Castellano e Inglés
- Manuales: 400, de Juegos y/o Utilitarios
- Accesorios: Fuentes, Reset, Interfases

para grabador, cajas portadiskettes, fundas, cortadores para Disk, Joystick, Fast Load, Cartridge especiales, etc.

ENVIOS AL INTERIOR S/CARGO

Horario 15 a 20 hs.

MEGASOFT — Av. Cabildo 2967 L. 15 T.E.: 701-2569

ELECTROSOUND

CONVERSION PAL-N EN TS 2068
CARTRIDGE EMULADOR SPECTRUM
MAGIC LOADER Y MAGIC COPI

VIAMONTE 1336 PISO 8º of. 48 TE.: 45-8585 - CAP.

COMMODORE 64 - ATARI
REFORMAS A PAL - N

Servicio Técnico - Monitores
Personal Computer - Periféricos - Video Juegos
Conversión de TV a BI-NORMA

zapata 586 (alt. cabildo al 600) tel. 553-1740

MICROCOMPUTADORAS
NUEVAS Y USADAS

COMPRA - VENTA - CANJE
- JOYSTICKS - CASSETTES - DISKETTES - LIBROS
CURSOS DE COMPUTACION NIÑOS Y ADULTOS

URIBURU 291 **DATA CLUB** TE: 45-3999 46-5817

SERVICE DYPEA

ELECTRONICA DE ALTO NIVEL
- COMPUTADORAS • DISKETTERAS
- VIDEO GAMES • IMPRESORAS
- VIDEOCASSETERAS • PAL N/NTSC

PASO 753 - TE.: 47-5337

CONVERSION GRABADORES PARA COMMODORE

Convertimos su grabador común (funcione o no) en DATASET (igual grabador Commodore) No es la problemática interfase, olvídela y reemplácela. Convénzace: Commodore no vende interfases. Si busca una solución definitiva y económica, llámenos. Compatible con turbotape y turboplus. Además SERVICE Consolas, Disketeras y Dataset. Programas.

Zonas disponibles en el interior para representantes
(Más información por carta a Victor Martinez 376 - (1406) - Cap. Fed.)

CAPITAL Y GRAN BS. AS. - 432-9925 - 941-5101

NOVEDAD
RTTY PARA TS 2068

INTERFAZ MODEM TRANSMISION y RECEPCION DE RTTY, BAUDOT y ASCII, 45 A 600 BAUDIOS SHIFT y TONOS VARIABLES, BUFFER DE RECEPCION y TRANSMISION, MODULO COMPACTO, ALIMENTACION DESDE LA COMPUTADORA, ETC. OFERTA ₳ 79

GALICIA 1279 - 1º "B" TE. 611-0505 **ENVIOS AL INTERIOR**

---

commodore 64

EL "SOFT" EN CASSETTE TE OFRECE
TODOS LOS TITULOS DE LOS MEJORES JUEGOS
EN CASSETTE - NADIE TIENE TANTO
STOCK EN JUEGOS Y UTILITARIOS
TE: 38-9678
MONTEVIDEO 31 5º 7 - CAPITAL
ENVIOS AL INTERIOR

Mr BYTE
SOFTWARE

SECI soft PROGRAMAS

PARA SU commodore 64
en diskettes o cassettes

781-6538 - LA PAMPA 2041 L.3

JUEGOS PARA COMMODORE 64

Al mejor precio de plaza

1 CASSETTE C/40 JUEGOS
Por solo ¡¡¡₳ 7.50!!!

Hay gran variedad

Lunes a Viernes de 10 a 13 y 15 a 19 Hs. Sab. De 10 a 14 hs.
CORDOBA 4495 (1 cuadra de Canning)
ESMERALDA 486 7º K (esq. Lavalle)

PROGRAMAS

DEMOSTRACION DE BIT-MAP



COMP.: TI99/4A
CLAS.: EDU



Utilicemos este programa en Assembler para graficar en nuestra TI,, las instrucciones aparecerán por nuestra pantalla.

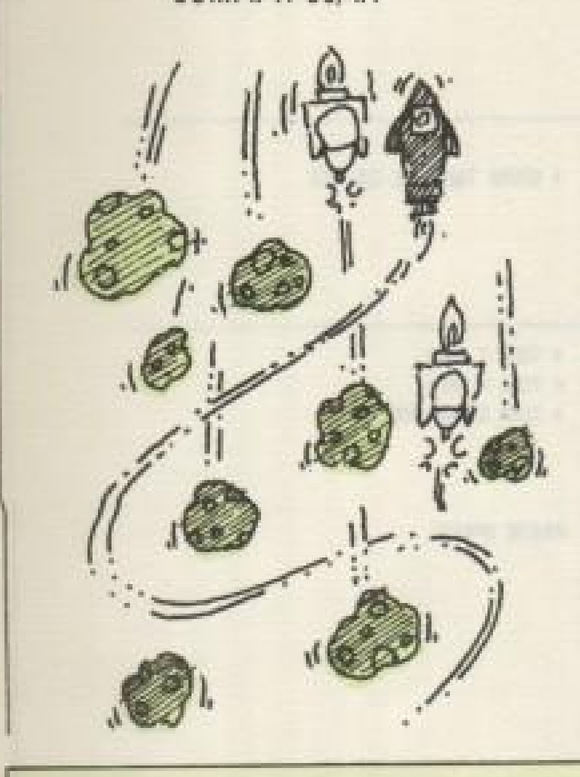
```
* DEMOSTRACION DE BIT-MAP.
* ---------------------
* Este es un simple programa desarrolado principalmente para aque-
* llos que desean explotar al maximo las posibilidades graficas de
* su TI-99. Aprendiendo a manejar el poderoso modo grafico,Bit-Map.
* En este modo su TI le permite definir 768 carracteres distintos,
* y darle a cada uno de ellos hasta 16 colores distintos.
* Debido a la cantidad de memoria necesaria para esto, es que solo
* podemos acceder a este modo grafico con programas en Assembler,
* ya que la VDP sera casi completamente usada para video.
* Por lo tanto no podemos acceder a este modo desde basic.
* El programa en Assembler presentado aqui esta realizado en forma
* didactica, por lo que para los mas adelantados tal vez puedan
* resumirlo considerablamente.
* Para mayor informacion referirse al manual del Editor Assembler.
*
*+-----------------------------------------------------------+
*!                                                           !
*!                                                           !
*!                                                           !
*!          DEMOSTRACION DEL MODO GRAFICO BIT-MAP            !
*!          ---------------------------------                !
*!                                                           !
*!          por Alberto de Harenne. 18/11/85.                !
*+-----------------------------------------------------------+
*! el cursor se mueve con las teclas <E S D X W R Z C>       !
*! <FCTN 4> borra la pantalla                                !
*! <ENTER> cambia color del trazo.                           !
*+-----------------------------------------------------------+
       DEF  DEMO
       REF  VWTR,VSBW,VSBR,VMBW,KSCAN
*------------------------------------------------------------
BLAN   DATA 0,0,0,0
SPR    DATA >6080,>000F,>D000       *DEFINICION DEL SPRITE
CAR    DATA >FCFC,>F8FC,>FEDF,>0F07 *CARACTER DEL SPRITE
ENTER  BYTE 13
FCTN4  BYTE 2
E      BYTE 69
X      BYTE 88
S      BYTE 83
D      BYTE 68
W      BYTE 87
R      BYTE 82
Z      BYTE 90
C      BYTE 67
*------------------------------------------------------------
DEMO   LWPI >8300
       BLWP @BITMAP     *AJUSTA LOS REGISTROS DE VIDEO A BIT-MAP
       BLWP @INIT       *INICIALIZA LA VDP PARA BIT-MAP.
       LI   R6,>80      * X INICIAL
       LI   R7,>60      * Y INICIAL
       LI   R8,15       * COLOR INICIAL
*------------------------------------------------------------
       LI   R0,>3800
       LI   R1,CAR      *DEFINE CARACTER DEL SPRITE
       LI   R2,8
       BLWP @VMBW
*------------------------
       LI   R0,>1B00
       LI   R1,SPR
       LI   R2,5        *DEFINE SPRITE
       BLWP @VMBW
*------------------------------------------------------------
       LI   R5,10
TECLA  CLR  @>837C
       CLR  @>8374
       BLWP @KSCAN
       LIMI 2
       LIMI 0
       DEC  R5
       JNE  TECLA
       LI   R5,10
       CB   @>8375,@E
       JNE  CX
       BL   @ARRIBA
       BLWP @PLOT
       JMP  TECLA
CX     CB   @>8375,@X
       JNE  CS
       BL   @ABAJO
       BLWP @PLOT
       JMP  TECLA
CS     CB   @>8375,@S
       JNE  CD
       BL   @IZQU
       BLWP @PLOT
       JMP  TECLA
CD     CB   @>8375,@D
       JNE  CW
       BL   @DERE
       BLWP @PLOT
       JMP  TECLA
CW     CB   @>8375,@W
       JNE  CR
       BL   @ARRIBA
       BL   @IZQU
       BLWP @PLOT
       JMP  TECLA
CR     CB   @>8375,@R
       JNE  CZ
       BL   @ARRIBA
       BL   @DERE
       BLWP @PLOT
       JMP  TECLA
CZ     CB   @>8375,@Z
       JNE  CC
       BL   @ABAJO
       BL   @IZQU
       BLWP @PLOT
       JMP  TECLA
CC     CB   @>8375,@C
       JNE  CCOLOR
       BL   @ABAJO
       BL   @DERE
       BLWP @PLOT
       JMP  TECLA
CCOLOR CB   @>8375,@ENTER
       JNE  CF4
       BL   @COLOR
       JMP  TECLA
CF4    CB   @>8375,@FCTN4
       JNE  TECLA
       BLWP @INIT
       JMP  TECLA
*------------------------
ABAJO  CI   R7,191
       JNE  AB1
       RT
```

```
AB1      INC  R7
         RT
*-----------------------
ARRIBA CI   R7,0
         JNE  AR1
         RT
AR1      DEC  R7
         RT
*-----------------------
IZQU     CI   R6,0
         JNE  IZ1
         RT
IZ1      DEC  R6
         RT
*-----------------------
DERE     CI   R6,255
         JNE  DE1
         RT
DE1      INC  R6
         RT
*-----------------------------------------------
COLOR    CI   R8,15
         JNE  CO1          *CAMBIA COLOR DEL TRAZO
         LI   R8,1
         JMP  CO2
CO1      INC  R8
CO2      MOV  R8,R1
         SLA  R1,8
         LI   R0,>1803     *CAMBIA COLOR DE SPRITE
         BLWP @VSBW
         RT
*-----------------------------------------------
REG1     BSS  32
BITMAP DATA REG1,BIT
*-----------------------
REG2     BSS  32
INIT     DATA REG2,INI
*-----------------------
REG3     BSS  32
PLOT     DATA REG3,PLO
*-----------------------------------------------
BIT      LI   R0,>0002
         BLWP @VWTR        * AJUSTA VDPRO PARA BIT MAP
         LI   R0,>0206
         BLWP @VWTR        * CAMBIA TABLA DE PANTALLA A >1800
         LI   R0,>0403
         BLWP @VWTR        * CAMBIA TABLA DE DEF. DE CARAC. A >0000
         LI   R0,>03FF
         BLWP @VWTR        * CAMBIA TABLA DE COLORES A >2000
         LI   R0,>0701
         BLWP @VWTR        * COLOR DE PANTALLA NEGRO
         LI   R0,>0536
         BLWP @VWTR        * CAMBIA TABLA DE SPRITES A >1C00
         LI   R0,>0607
         BLWP @VWTR        * DEF.DE CARAC.DE SPRITES A >3800
         RTWP
*-----------------------------------------------
INI      LI   R0,>1800
         CLR  R1
         LI   R2,3
A1       SWPB R1           * CREA LA TABLA DE PANTALLA
         BLWP @VSBW
         SWPB R1
         INC  R0
         INC  R1
         CI   R1,256
         JLT  A1
         CLR  R1
         DEC  R2
         JNE  A1
*-----------------------------------------------
         CLR  R0
         LI   R1,BLAN
         LI   R2,8         * BORRA TABLA DEFINICION DE CARACTERES
A2       BLWP @VMBW
         AI   R0,8
         CI   R0,>1800
         JLT  A2
*-----------------------------------------------
         LI   R0,>2000
A3       BLWP @VMBW        * BORRA TABLA DE COLORES
         AI   R0,8
         CI   R0,>3800
         JLT  A3
         RTWP
*-----------------------------------------------
PLO      MOV  @12(13),R6   * TOMA X (R6)
         MOV  @14(13),R7   * TOMA Y (R7)
         MOV  @16(13),R8   * TOMA COLOR (R8)
         MOV  R7,R1
         SLA  R1,8
         LI   R0,>1800
         BLWP @VSBW        *MUEVE SPRITE
         MOV  R6,R1
         SLA  R1,8
         INC  R0
         BLWP @VSBW
         MOV  R7,R4
         SLA  R4,5
         SOC  R7,R4
         ANDI R4,>FF07
         MOV  R6,R5
         ANDI R5,7         * DEJA EN R5 EL BIT A PRENDER.
         A    R6,R4
         S    R5,R4
         MOV  R4,R0        * DEJA EN R0 EL BYTE A MODIFICAR.
*-----------------------------------------------
         BLWP @VSBR
         MOVB @MASC(R5),R3
         SOC  R3,R1
         BLWP @VSBW
         AI   R0,>2000
         MOV  R8,R1
         SLA  R1,12
         BLWP @VSBW
         RTWP
MASC     DATA >8040,>2010
         DATA >0804,>0201
         END
```

RUTA SOLAR



5º PREMIO DEL 3er. CONCURSO K64
AUTOR: Fabio Rossi
COMP.: TI 99/4A





Tengo 16 años y escribí este juego para TI99/4A, con módulo de extended basic.

Consiste en evitar chocar con rocas, naves y monstruos que avanzan hacia nuestra nave. Una vez que pasamos estas dificultades llegamos a una base donde se cambiará la nave por otra de distinta definición. Cada base que pasemos aumentará la velocidad de las rocas.

Otro obstáculo es el combustible, cuyo marcador se encuentra en la parte inferior izquierda de la pantalla. Si éste marca vacío (aguja hacia la izquierda) la nave se viene a pique. Para evitar esto podemos llenarlo "chocando" con el tanque azul que pasa por el piso.

Además es imposible dejar la nave quieta, ya que si no el jugador podría buscar una posición en la cual no chocaría nunca.

A la derecha del marcador de combustible hay una pequeña pantalla que nos avisa la proximidad de las naves enemigas y de las bases.

En el centro de la primer fila se marca el record. Si éste es superado se pasa a otra pantalla, en la cual se le pide el nombre al jugador y se le toca la canción "Oh Susana".

Cuando aparece la presentación hay que apretar una tecla para que cese el ruido de sirena y la computadora pida si desea instrucciones.

```
10 REM ***************
20 REM ***************
30 REM **RUTA SOLAR**
40 REM ***************
50 REM ***************
60 REM
70 REM
80 REM POR FABIO D. ROSSI
90 REM
100 REM NOVIEMBRE 1985
110 REM
120 REM PARA TI99/4A
130 REM
140 REM TI EXTENDED BASIC
150 REM
160 REM **PRESENTACION**
170 RECORD$="TI99/4A" :: RECORD=0 :: X=1 :: CALL CLEAR :: CALL SCREEN(2):: A=3
180 CALL CHAR(40,"01001F003F007F008000F800FC00FE00FF007F003F000700FF00FE00FC00E
0")
190 CALL CHAR(44,"FF00FF00FF00FF000F000F000F000F00F000F000F000F000")
200 DISPLAY AT(6,4):"   (,,) ,  , ,,,, (,,)" :: DISPLAY AT(13,4):"(,,, (,,) ,
 (,,) (,,)"
210 DISPLAY AT(7,4):"    , , , ,  -.  ,  ," :: DISPLAY AT(14,4):",      , , ,
 ,  , , ,"
220 DISPLAY AT(8,4):"   ,,,+ , ,  -.  ,,,," :: DISPLAY AT(15,4):"*,,) ,  , ,
 ,,,, ,,,+"
230 DISPLAY AT(9,4):"    , ,  , ,  -.  ,  ," :: DISPLAY AT(16,4):"     , ,  , ,
 ,  , , ,"
240 DISPLAY AT(10,4):"    , ,, *,,+  -.  ,  ," :: DISPLAY AT(17,4):",,,+ *,,+ *,,
, ,   , , ,," :: Z=800
250 CALL KEY(0,K,S):: GOSUB 290 :: CALL SOUND(-199,Z,5):: Z=Z+15 :: IF Z>1000 TH
EN 270 :: IF S=0 THEN 250
260 GOTO 310
270 CALL KEY(0,K,S):: GOSUB 290 :: CALL SOUND(-199,Z,5):: Z=Z-20 :: IF Z<700 THE
N 250 :: IF S=0 THEN 270
280 GOTO 310
```

```
290 CALL COLOR(2,A,2):: A=A+1 :: IF A=17 THEN 300 :: RETURN
300 A=3 :: RETURN
310 FOR B=1 TO 8 :: CALL COLOR(B,A,2):: NEXT B :: CALL COLOR(12,A,2)
320 ME$="                          RUTA SOLAR POR FABIO ROSSI PARA TI99/4A
   DESEA INSTRUCCIONES (S/N)?                               "
330 FIL=24 :: GOSUB 1700
340 CALL KEY(0,K,S):: IF S=0 THEN 340 :: IF K=83 THEN 360 :: IF K=78 THEN 470 EL
SE 340
350 FOR A=0 TO 30 STEP 7 :: CALL SOUND(-99,1000,A,-1,A):: NEXT A
360 ME$="                         UD DEBERA MANEJAR SU NAVE EVITANDO CHOCAR CO
N LAS ROCAS Y NAVES KAMIKASES                  "
370 FIL=24 :: GOSUB 1700
380 ME$="                       EN EL BORDE INFERIOR IZQUIERDO DE SU PANTALLA
 ESTA EL MARCADOR DE COMBUSTIBLE                "
390 FIL=24 :: GOSUB 1700
400 ME$="                         PARA LLENARLO DEBERA CHOCAR EL TANQUE AZUL
QUE PASA POR EL PISO                          "
410 FIL=24 :: GOSUB 1700
420 ME$="                         EN CADA NIVEL QUE AVANCE SE LE CAMBIARA LA
NAVE                               " :: FIL=24 :: GOSUB 1700
430 ME$="                         UD TIENE 5 NAVES / MANEJELAS CON LAS TECLAS
 E y X                             " :: FIL=24 :: GOSUB 1700
440 ME$="                         PRESIONE UNA TECLA PARA COMENZAR        BU
ENA SUERTE!      "
450 FIL=24 :: GOSUB 1700
460 CALL KEY(0,K,S):: IF S=0 THEN 460
470 FOR A=0 TO 30 STEP 7 :: CALL SOUND(-99,1000,A,-1,A):: NEXT A
480 NAVE=96 :: CALL MAGNIFY(3):: RESTORE 490 :: FOR HG=1 TO 10 :: READ BASE$(HG)
:: NEXT HG
490 DATA "LITIO","RUBIDIO","TITANIO","LANTANIO","ACTINIO","TANTALIO","CERIO","TO
```

```
RIO","WOLFRAMIO","TECNECIO"
500 NIVEL=1 :: VIDA=5 :: T=0 :: NAVE=96 :: ENE=116 :: PUNTO=0 :: CALL CLEAR :: 
ALL SCREEN(2)
510 REM **PISO BASE**
520 CALL CHAR(86,"CCCC3333CCCC3333")
530 REM**COSO BASE**
540 CALL CHAR(87,"FF81BDA5A5BD81FF")
550 REM **FONDO**
560 CALL CHAR(40,"")
570 REM **COMB**
580 CALL CHAR(88,"0303030303030303030000000000000000000000000000000")
590 REM**EXPLOSION**
600 CALL CHAR(36,"000400110101 4F0C0028062400442000000 00A200869704800A8022004872
8")
610 REM **HOMBRE 1**
620 CALL CHAR(40,"1C1C1C1C0808182C2A29081416724203000000000000000000")
630 REM **HOMBRE 2**
640 CALL CHAR(44,"1C1C1C1C080808181C1808080C7444060000000000000000000")
650 REM **HOMBRE 3**
660 CALL CHAR(60,"1C1C1C1C080808182C2A080C0A1C180C0000000000000000000")
670 REM **HOMBRE 4**
680 CALL CHAR(92,"1C1C1C1C0808081818 1C0A081C163222300000000000000000000")
690 REM **AVION 1**
700 CALL CHAR(96,"000000000000000000006060734FFE0F000000000000000000000000C0FC
F8")
710 REM **PLATO 1**
720 CALL CHAR(104,"0303030101010F0F0FFFE6667F3F1E0780808000 0080F0F0F0FF6766FEF
8E")
730 REM **PLATO 2**
740 CALL CHAR(108,"000000000000000010307070FFF7F3720000000000000000 80C0E0E0F0FFF
C04")
750 REM **AVION 2**
760 CALL CHAR(100,"0000000000000000E0783CFE7F3FFF1F000000000000000000000000080F
FC")
770 REM **NAVE**
780 CALL CHAR(112,"00000000071F080402FF3F0204081F070000000080F0204080FCFF8040
080")
790 REM ***ROCA***
800 CALL CHAR(116,"0103061C3163464C40721A0A080F0100C0603C06561434A6624246D43CE
810 REM **FLECHA 1**
820 CALL CHAR(120,"00000000000007FC7C070000000000000000000073CE08080E03C07")
830 REM **FLECHA 2**
840 CALL CHAR(124,"000000000C183871DF7138180C000000000000003060E2CE7BCEE2603")
850 REM **MONSTRUO**
860 CALL CHAR(128,"04041C147C3C6C0E0E0F070100000000004060707078787B72FEFCFC7")
870 REM **TANQUE**
880 CALL CHAR(132,"0000031E3D3D3E1F0B0A1A16342420000000C078FCFC7CF8D05058682C2
4")
890 REM **ARBOL**
900 CALL CHAR(136,"E0BDE507715171213D07E3A3E34F790100073DA7E0809C949C88E838008
080")
910 REM **EDIFICIO**
920 CALL CHAR(140,"FF9999FF9999FF9999FF9999FFFFC4C4000000000000000000FFFF9999FFF
747")
930 REM ***COMIENZO***
940 CALL CLEAR :: CALL COLOR(2,16,16):: CALL SCREEN(2):: FOR A=1 TO 18 :: CALL
CHAR(A,1,40,32):: NEXT A
950 FOR A=3 TO 8 :: CALL COLOR(A,7,16):: NEXT A :: FOR A=21 TO 23 :: CALL HCHA
A,14,40,16):: NEXT A
960 PA$=STR$(RECORD)&"("&RECORD$ :: VOL=30 :: FIL=1 :: COL=10 :: GOSUB 1670
970 PA$="COMBUSTIBL" :: FIL=20 :: COL=1 :: VOL=30 :: GOSUB 1670 :: CALL HCHAR(
,2,40,10)
980 PA$="PUNTOS:" :: FIL=23 :: COL=1 :: VOL=30 :: GOSUF 1670 :: DISPLAY AT(24,
:PUNTO
990 CALL SPRITE(£2,NAVE,7,73,10,0,0)
1000 CALL SPRITE(£3,132,8,130,50,0,-10-NIVEL,£4,136,7,130,140,0,-10-NIVEL,£5,1
,12,130,170,0,-10-NIVEL)
1010 CALL SPRITE(£1,88,7,160,82):: CALL SPRITE(£12,88,2,160,1):: CALL SPRITE(£
```




```
,88,2,160,97)
1020 FOR D=1 TO 2 :: PA$="ATENCION" :: FIL=22 :: VOL=0 :: COL=17 :: GOSUB 167
1030 FOR A=1 TO 10 :: NEXT A
1040 PA$="((((((((" :: FIL=22 :: VOL=0 :: COL=17 :: GOSUB 1670 :: FOR A=1 TO
 :: NEXT A :: NEXT D
1050 V=1 :: FOR A=6 TO 11 :: C=INT(RND*9)+3 :: VEL=(RND*10):: V=V+18 :: CALL
ITE(£A,ENE,C,V,240,0,-VEL-(NIVEL*1.5)-5):: NEXT A
1060 CALL KEY(0,X,S):: T=T+1 :: IF T>300 THEN 1250 :: CALL MOTION(£1,0,-1)::
L DISTANCE(£2,£12,D):: IF D>22000 THEN 1120
1070 IF D<1000 THEN 1150
1080 IF X=69 THEN 1090 :: IF X=88 THEN 1120 :: CALL COINC(ALL,C):: T=T+1 :: I
=0 THEN 1060 ELSE 1150
1090 CALL MOTION(£2,-10,0):: CALL SOUND(-250,-5,25):: CALL MOTION(£1,0,0)
1100 CALL COINC(ALL,C):: T=T+1 :: IF C=0 THEN 1060 ELSE 1150
1110 GOTO 1060
1120 CALL MOTION(£2,10,0):: CALL SOUND(-250,-6,25):: CALL MOTION(£1,0,0)
1130 CALL COINC(ALL,V):: IF V=0 THEN 1060 ELSE 1150
1140 GOTO 1060
1150 CALL DISTANCE(£12,£1,A):: IF A<18 THEN CALL MOTION(£2,14,0)ELSE 1170
1160 CALL DISTANCE(£2,£12,D):: IF D<1500 THEN 1170 :: CALL SOUND(99,(D/100)+1
10):: GOTO 1160
1170 CALL DISTANCE(£2,£3,D):: IF D<170 THEN 1200 :: CALL SOUND(1000,-7,2,110,
:: CALL MOTION(£2,0,0)
1180 CALL PATTERN(£2,36):: VIDA=VIDA-1 :: FOR D=1 TO 10 :: CALL COLOR(£2,15):
OR A=1 TO 10 :: NEXT A :: CALL COLOR(£2,7):: FOR A=1 TO 10
1190 NEXT A :: NEXT D :: FOR A=1 TO 100 :: NEXT A :: CALL DELSPRITE(ALL):: IF
DA=0 THEN 1720 :: GOTO 990
1200 A=200 :: CALL MOTION(£2,0,0,£3,0,0,£4,0,0,£5,0,0):: CALL MOTION(£1,0,6)
1210 CALL SOUND(-100,A,5):: A=A+5 :: CALL COINC(£1,£13,16,D):: IF D=0 THEN 12
:: CALL MOTION(£1,0,0):: CALL DELSPRITE(£2)
1220 CALL SPRITE(£2,NAVE,7,116,10):: FOR A=0 TO 30 STEP 5 :: CALL SOUND(-99,1
,A,900,A):: NEXT A
1230 CALL MOTION(£3,0,-10):: CALL MOTION(£4,0,-10):: CALL MOTION(£5,0,-10)::
L MOTION(£1,0,-1):: PUNTO=PUNTO+10
1240 DISPLAY AT(24,1):PUNTO :: GOTO 1060
1250 ENE=ENE+4 :: T=0 :: CALL MOTION(£2,0,0):: FOR A=6 TO 11 :: CALL DELSPRIT
A)
1260 FOR D=1 TO 20 STEP 7 :: CALL SOUND(99,(A*100)+300,D):: NEXT D :: NEXT A
CALL MOTION(£1,0,0):: FOR A=1 TO 500 :: NEXT A
1270 IF ENE=132 THEN 1280 :: PUNTO=PUNTO+100 :: DISPLAY AT(24,1):PUNTO :: GOT
010
1280 REM **BASE**
1290 ON NIVEL GOTO 1300,1310,1320,1330,1340,1350,1360,1370,1380,1390
1300 P=4 :: F=16 :: GOTO 1400
1310 P=5 :: F=6 :: GOTO 1400
1320 P=6 :: F=8 :: GOTO 1400
1330 P=9 :: F=16 :: GOTO 1400
1340 P=13 :: F=12 :: GOTO 1400
1350 P=13 :: F=16 :: GOTO 1400
1360 P=14 :: F=15 :: GOTO 1400
1370 P=15 :: F=5 :: GOTO 1400
1380 P=16 :: F=6 :: GOTO 1400
1390 P=16 :: F=15 :: GOTO 1400
1400 PA$="ATENCION" :: FIL=22 :: VOL=0 :: COL=17 :: GOSUB 1670
1410 FOR A=1 TO 10 :: NEXT A
1420 PA$="((((((((" :: FIL=22 :: VOL=0 :: COL=17 :: GOSUB 1670
1430 PA$="BASE("&BASE$(NIVEL):: VOL=0 :: FIL=22 :: COL=13 :: GOSUB 1670
1440 FOR A=1 TO 10 :: NEXT A
1450 PA$="(((((((((((((((" :: VOL=0 :: FIL=22 :: COL=13 :: GOSUB 1670
1460 CALL MOTION(£3,0,0):: CALL MOTION(£4,0,0):: CALL MOTION(£5,0,0):: CALL M
ON(£2,0,30):: CALL SOUND(1500,-6,10)
1470 FOR A=1 TO 300 :: NEXT A
1480 CALL DELSPRITE(ALL):: CALL CLEAR :: CALL SCREEN(F):: FOR A=1 TO 7 :: CAL
OLOR(A,P,F):: NEXT A :: CALL COLOR(8,P,P)
1490 B=10 :: C=12 :: CALL HCHAR(5,1,86,32):: FOR A=6 TO 10 :: B=B+1 :: C=C-2
CALL HCHAR(A,B,87,C):: NEXT A :: CALL HCHAR(18,1,90,32)
1500 IF NAVE=112 THEN 1510 :: CALL SPRITE(£1,NAVE+4,7,122,172):: GOTO 1520
1510 CALL SPRITE(£1,96,7,122,172)
```

```
1520 CALL SPRITE(£2,NAVE,7,100,1,0,10):: CALL SOUND(1600,-6,20):: FOR A=1 TO 300
 :: NEXT A :: CALL SOUND(1300,-7,20):: CALL MOTION(£2,5,0)
1530 FOR A=1 TO 210 :: NEXT A :: CALL MOTION(£2,0,0):: FOR A=1 TO 300 :: NEXT A
1540 FOR A=1 TO 10 :: CALL SPRITE(£3,92,16,121,73):: CALL SOUND(10,2000,10,-3,10
):: CALL COLOR(£3,2):: CALL SOUND(10,3000,10,-3,10):: NEXT A
1550 FOR A=1 TO 100 :: NEXT A :: CALL MOTION(£3,0,4):: FOR A=1 TO 13 :: CALL PAT
TERN(£3,40):: CALL COLOR(7,P,F):: FOR D=1 TO 20 :: NEXT D
1560 CALL PATTERN(£3,44):: CALL COLOR(7,F,P):: FOR D=1 TO 20 :: NEXT D
1570 CALL PATTERN(£3,60):: CALL COLOR(7,P,F):: FOR D=1 TO 20 :: NEXT D
1580 CALL PATTERN(£3,92):: CALL COLOR(7,F,P):: FOR D=1 TO 20 :: NEXT D :: NEXT A
 :: FOR A=1 TO 100 :: NEXT A
1590 CALL MOTION(£3,0,0):: FOR S=1 TO 100 :: NEXT S :: FOR A=1 TO 20 :: CALL COL
OR(£3,16):: CALL COLOR(£3,2):: NEXT A :: CALL DELSPRITE(£3)
1600 FOR A=0 TO 30 STEP 5 :: CALL SOUND(99,1700,A,-1,A):: NEXT A
1610 FOR A=1 TO 200 :: NEXT A :: FOR E=1 TO 10 :: PUNTO=PUNTO+50 :: DISPLAY AT(2
4,1):PUNTO :: CALL SOUND(25,1000,0):: CALL SOUND(25,1000,5):: NEXT E
1620 CALL SOUND(1300,-7,10):: CALL MOTION(£1,-5,0):: FOR A=1 TO 280 :: NEXT A
1630 CALL SOUND(2200,-6,10):: CALL MOTION(£1,0,10):: FOR A=1 TO 420 :: NEXT A ::
 CALL DELSPRITE(£1):: FOR A=1 TO 300 :: NEXT A
1640 IF NIVEL=10 THEN 1650 :: NIVEL=NIVEL+1
1650 NAVE=NAVE+4 :: IF NAVE=116 THEN NAVE=96 :: T=0
1660 ENE=116 :: CALL DELSPRITE(ALL):: CALL CLEAR :: CALL CHARSET :: GOTO 510
1670 REM **IMPRESION**
1680 FOR PA=1 TO LEN(PA$):: CALL HCHAR(FIL,COL+PA,ASC(SEG$(PA$,PA,1))):: CALL SO
UND(-100,(PA*20)+800,VOL):: NEXT PA
1690 RETURN
1700 REM***LETRADO***
1710 FOR ME=1 TO LEN(ME$)-27 :: DISPLAY AT(FIL,1):SEG$(ME$,ME,28):: FOR A=1 TO 1
0 :: NEXT A :: NEXT ME :: RETURN
1720 IF PUNTO<RECORD THEN 1920 :: CALL CLEAR :: CALL CHARSET :: CALL SCREEN(8)
1730 FOR A=1 TO 8 :: CALL COLOR(A,14,8):: NEXT A
1740 M$="81C3663C183C66C3" :: CALL CHAR(32,M$):: CALL CHAR(40,M$)
1750 PA$="FELICITACIONES!!" :: FIL=3 :: VOL=30 :: COL=7 :: GOSUB 1670
1760 PA$="OBTUVISTE UN" :: FIL=6 :: COL=9 :: VOL=30 :: GOSUB 1670
1770 PA$="NUEVO RECORD" :: FIL=9 :: COL=9 :: VOL=30 :: GOSUB 1670
1780 PA$="ESCRIBE TU NOMBRE" :: FIL=12 :: COL=6 :: VOL=30 :: GOSUB 1670
1790 ACCEPT AT(20,7)SIZE(10):RECORD$ :: RECORD=PUNTO
1800 REM**CANCION**
1810 RESTORE 1820 :: FOR A=1 TO 57 :: READ DUR,NOTA :: CALL SOUND(DUR,NOTA,5)::
NEXT A
1820 DATA 100,349,100,392,200,440,200,523,300,523,100,587,200,523,200,440,300,34
9,100,392
1830 DATA 200,440,200,440,200,392,200,349,400,392,200,392,100,349,100,392,200,44
0,200,523
1840 DATA 300,523,100,587,200,523,200,440,300,349,100,392,200,440,200,440,200,39
2,200,392
1850 DATA 800,349,400,494,400,494,200,587,400,587,200,587,200,523,200,523,200,44
0,200,349
1860 DATA 400,392,200,392,100,349,100,392,200,440,200,523,300,523,100,587,200,52
3,200,440
1870 DATA 300,349,100,392,200,440,200,440,200,392,200,392,300,349
1880 PA$="DESEAS JUGAR DE NUEVO (S/N)?" :: FIL=24 :: COL=2 :: VOL=30 :: GOSUB 16
70
1890 CALL KEY(0,K,S):: IF S=0 THEN 1890 :: IF K=83 THEN 1900 :: IF K=78 THEN 191
0
1900 CALL CLEAR :: CALL CHARSET :: GOTO 480
1910 CALL CLEAR :: PRINT "             CHAU !!" :: FOR A=1 TO 10 :: PRINT :: CALL S
OUND(400,233,10,262,20):: CALL SOUND(10,1000,30):: NEXT A :: END
1920 CALL CLEAR :: CALL CHARSET :: CALL SCREEN(12):: FOR A=1 TO 8 :: CALL COLOR(
A,13,12):: NEXT A
1930 ME$="                             JUEGO TERMINADO              " :: FIL=10 :: GOSUB
1700
1940 ME$="                             DESEA JUGAR DE NUEVO (S/N)? " :: FIL=13 ::
GOSUB 1700
1950 CALL KEY(0,K,S):: IF S=0 THEN 1950
1960 FOR A=0 TO 30 STEP 5 :: CALL SOUND(-99,1000,A,-1,A):: NEXT A :: IF K=83 THE
N 1970 :: IF K=78 THEN 1910
1970 CALL CHARSET :: GOTO 480
```

SELECCIONAMOS

EL PROGRAMADOR DEL AÑO '86



COMPUTACION PARA TODOS

1º PREMIO
EL LINGOTE DE ORO DE K-64
(100 grs. Valor aprox. Bco. Municipal ₳ 1.000.-)



2º PREMIO: Una Consola 48 K
3º PREMIO: Una mesa para computador

BASES PARA PARTICIPAR EN EL CERTAMEN

Las bases y condiciones generales son las siguientes:

Una vez terminado y revisado tu programa, deberás enviarlo a la editorial grabado en un cassette o diskette, varias veces para mayor seguridad. (Inclusive grabado con dos grabadores distintos). Indicar en el cassette o diskette, los datos del programa, computadora y autor.
Otra condición es que sea original e inédito, es decir que no haya sido enviado a ninguna otra publicación. Si bien es preferible que vaya acompañado del listado del mismo por impresora, este no es imprescindible.
El programa deberá venir con un texto que aclare cuál es su nombre, objetivo, modo de uso, y explicación de cada una de sus partes, subrutinas y variables. Si posee lenguaje de máquina, es fundamental una buena explicación sobre su funcionamiento e ingreso a la máquina. No olvidarse los datos completos del autor o autores.
El texto se presentará en hojas tipo oficio y mecanografiado a doble espacio. No importa que la redacción no sea muy clara, eso queda por nuestra cuenta.

Jurado: Un jurado propio compuesto por profesionales en computación y usuarios de computadores, decidirá los resultados del mismo.
El criterio de elección, como siempre, se basará en originalidad de la idea; método de programación; efectos gráficos y sonoros; documentación del programa; presentación y ahorro de memoria. En la clase de programas del tipo no-juegos, se evaluará también la facilidad de manejo y explicación de los conceptos vertidos en el programa.

Cierre: El cierre de recepción de trabajos para concurso de programas será el: 31/07/86. (K64 se reserva el derecho de publicación de los programas recibidos, como asimismo la devolución del material).

Sorteo Mensual: Todos los meses se sortearán 20 cassettes entre los programas recibidos.



CRIPTOGRAFIA



COMP.: DREAN COMMODORE 64
CLAS.: ENT



Valiéndonos de este programa podremos efectuar y descifrar mensajes en clave, utilizando la misma técnica que empleó Alemania en la segunda guerra mundial. La única limitación que presenta es que no podremos incluír números, símbolos ni espacios.

```
1 PG$="CRIPTOGRAFIA":AU$="ALEX BREED":A2
$="F.D.PINEIRO Y LEONE":BG$="RETURN"
3 REM COPYRYGHT (C) 1984 THE CODE WORKS
4 REM BOX 6905. SANTA BARBARA. CA 93160
5 REM AS OF 25JUL84
6 REM TRAD.& MOD.F.D.PINEIRO Y LEONE
10 REM AS OF 06OCT85
20 REM ***VARIABLES UTILIZADAS***
21 REM CT=NRO.MAX.DE LINEAS POR MEENSAJE
.*CT$()=ALMACENAMIENTO MJE.*DF$=INPUT RT

22 REM F=COD.ASCII D/CARACTS.EN CODIGO.*
G=CONT.ESPACIOS.*L=C.S.EN ELAB.CLAVE.
23 REM N=NRO.LINEAS EN MJE.*PR$=INPUT SE
CUENCIA FRACES.*S$()=DET.COD.EN ASCII.
24 REM S()=DET.CODIGO.*U$=FORMATEO PANTA
LLA.*W=RULO P/COD.*W$()=SUB-COD.EN ASCI
25 REM W()=SUB-COD.P/CADA COD.*X%=DIMENC
IONES P/CODIGOS Y SUB-CODIGOS.
30 REM ***MODO DE TRABAJO D/PROGRAMA***
31 REM 100-230:DATA Y DETERMINA PROCEDIM
IENTOS PARA DISTINTOS CODIGOS.
32 REM 240-260:INTERROGA SOBRE PROG.AUTO
MATICA.*270-450:PREPARA MAQ.P/USUARIO.
33 REM 460-500:MUESTRA CODIGOS USUARIO Y
 PIDE MENSAJE.*590-760*IMP.MJE.EN CLAVE
34 REM 770-800:PIDE OTRO MENSAJE CON MIS
MA PROGRAMAC.*890-900:IMPR.CODS.Y S-CODS

35 REM 910-940:RUTINA SI/NO.*950-980:APL
ICA CIFRADO C/CARACTER.
90 GOTO 62000
100 CT=50:DIM CT$(CT),X%(5,26)
110 POKE VIC+24,23
120 DATA 4,16,15,19,8,3,25,6,13,1,7,26,2
0,10,12,9,14,17,22,2,24,23,21,5,18,11
130 DATA 1,23,9,24,25,2,7,18,26,4,13,15,
20,17,10,8,5,3,14,22,16,12,21,11,6,19
140 DATA 1,8,2,20,3,14,4,19,5,26,6,17,7,
22,9,16,10,23,11,21,12,25,13,15,18,24
150 DATA 1,3,2,5,4,18,6,16,7,26,8,19,9,1
1,10,24,12,15,13,22,14,23,17,20,21,25
160 DATA 1,6,2,22,3,12,4,11,5,20,7,16,8,
24,9,13,10,17,14,25,15,26,18,23,19,21
170 X=RND(-TI)
171 PRINT"──▗▜▛─♠─▗♠:█CON ESTE PROGRAM
A USTED PO"
172 PRINT"DRA EFECTUAR Y DESCIFRAR MENSA
JES EN CLA"
173 PRINT"VE, USANDO LA MISMA TECNICA U
SADA POR ♠-"
174 PRINT"LEMANIA EN LA SEGUNDA GUERRA
MUNDIAL."
175 PRINT"                    LA UNICA LIMITACI
ON DE ESTE"
176 PRINT"SISTEMA RADICA EN LA IMPOSIBI
LIDAD DE CI-"
177 PRINT"FRAR NUMEROS, SIMBOLOS Y ESPA
CIOS.
178 PRINT"████████PARA COMENZAR PULSE CUA
LQUIER TECLA."
179 GET E$:IF E$=""THEN GOTO 179
180 PRINT"▔██/ \♠─ ♠..."
190 RESTORE
200 FOR W=1 TO 5
210 FOR K=1 TO 13:READ X,Y
220 X%(W,X)=Y:X%(W,Y)=X
230 NEXT K:NEXT W
240 PRINT"▔";
250 PR$="PROGRAMACION ♠UTOMATICA (S/N)":
DF$="N":GOSUB 910
260 IF LEFT$(IN$,1)="S"THEN GOSUB 810:GO
TO 460
270 T$="ABCDE"
280 FOR K=1 TO 3
290 PRINT"█─UAL DISCO EN RAHURA"K"(";T$;
")? ";:GOSUB 60000
300 IF LEN(IN$)<>1 THEN PRINT"█UNA LETRA
 POR FAVOR.":GOTO 290
310 IF IN$>="A"AND IN$<="E"THEN 330
320 PRINT"█UNA LETRA DE LA 'A' A LA 'E',
 POR FAVOR.":GOTO 290
330 T=ASC(IN$)-64
340 IF MID$(T$,T,1)=" "THEN PRINT"██ISC
O ";IN$;" YA ESTA OCUPADO.":GOTO 290
350 U$="":IF T>1 THEN U$=U$+LEFT$(T$,T-1
)
360 T$=U$+" "+MID$(T$,T+1)
370 W$(K)=IN$:W(K)=T
380 NEXT K
390 FOR K=1 TO 3
400 PRINT"█─ODIGO PARA DISCO"K"(A-Z)? ";

410 GOSUB 60000:IF LEN(IN$)<>1 THEN PRIN
T"█UNA LETRA POR FAVOR.":GOTO 400
420 T=ASC(IN$)-64
430 IF T<1 OR T>26 THEN PRINT"█UNA LETRA
████████           , POR FAVOR!":GOTO 400
440 S(K)=T:S$(K)=IN$
450 NEXT K
460 PRINT"███QUI ESTAN LOS DISCOS Y CODI
GOS:"
470 PRINT"█(UD.LOS NECESITA P/DESCIFRAR
EL MENSAJE)"
480 GOSUB 890
490 FOR K=1 TO 3:S(K)=ASC(S$(K))-64:NEXT
 K
500 PRINT"█POR FAVOR DEME SU MENSAJE."
510 PRINT"█INGRESE UNA LINEA EN BLANCO C
UANDO TERMINE."
520 FOR I=0 TO CT
530 PRINT"█? █";
540 GOSUB 60000:IF IN$=""THEN N=I-1:GOTO
 590
545 IF IN$="Q"THEN 60600
550 CT$(I)=IN$
560 NEXT I
570 PRINT"██ESO ES TODO LO QUE PUEDO MAN
IPULAR... PERDONE."
580 N=CT
590 PRINT"█";:IF N<0 THEN 770
600 K=0:G=0
610 PRINT"██EL MENSAJE CIFRADO ES....██"

620 FOR I=0 TO N:CT$=CT$(I)
630 FOR J=1 TO LEN(CT$)
640 F=ASC(MID$(CT$,J,1))-64:F=F AND 127
650 IF F<1 OR F>26 THEN 730
660 FOR W=-2 TO 2:L=3-ABS(W):GOSUB 950:N
EXT W:REM L=1,2,3,2,1
670 PRINT CHR$(F+64);
680 K=K+1:IF K>4 THEN PRINT" ";:K=0:G=G+
1:IF G>5 THEN PRINT:G=0
690 S(1)=S(1)+1:IF S(1)<27 THEN 730
700 S(1)=1:S(2)=S(2)+1:IF S(2)<27 THEN 7
30
710 S(2)=1:S(3)=S(3)+1:IF S(3)<27 THEN 7
30
720 S(3)=1
730 NEXTJ:NEXT I
750 IF K>0 THEN FOR J=1 TO 5-K:PRINTCHR$
(26*RND(1)+65);:NEXT
760 IF G>0 OR K>0 THEN PRINT
770 PR$="█OTRO MENSAJE (S/N)":DF$="S":GO
SUB 910
780 IF IN$="N"THEN 60600
790 PR$="IGUAL PROGRAMA DE MAQUINA (S/N)
":DF$="S":GOSUB 910:IF IN$="S"THEN490
800 GOTO 250
810 T$="ABCDE":FOR K=1 TO 3
820 T=INT(RND(1)*LEN(T$))+1
830 W$(K)=MID$(T$,T,1):W(K)=ASC(W$(K))-6
4
840 U$="":IF T>1 THEN U$=LEFT$(T$,T-1)
850 T$=U$+MID$(T$,T+1)
860 S(K)=INT(RND(1)*26)+1:S$(K)=CHR$(S(K
)+64)
870 NEXT K:RETURN
890 PRINT"██RANURA","DISCO","CODIGO":PRI
NT"█──────","──────","──────█"
900 FOR K=1 TO 3:PRINT K,W$(K),S$(K):NEX
T:PRINT"█";:RETURN
910 PRINT"█";PR$;"? ";:GOSUB 60000:IN$=L
EFT$(IN$,1):IF IN$=""THEN IN$=DF$
920 IF IN$="Q"THEN 60600
930 IF IN$="S" OR IN$="N"THEN RETURN
940 PRINT"█POR FAVOR RESPONDA 'SI' O 'NO
'.":GOTO 910
950 F=F-S(L)+1:IF F<1 THEN F=F+26
960 F=X%(W(L),F)
970 F=F+S(L)-1:IF F>26 THEN F=F-26
980 RETURN
60000 IN$="":ZT=TI:ZC=2:ZD$=CHR$(20)
60010 GET Z$:IF Z$<>"" THEN 60070
60020 IF ZT<=TI THEN PRINT MID$(" █",ZC,
1);"█";:ZC=3-ZC:ZT=TI+15
60030 GOTO 60010
60070 Z=ASC(Z$):ZL=LEN(IN$):IF(Z AND 127
)<32 THEN PRINT" █";:GOTO 60110
60090 IF ZL>=QI THEN 60010
60100 IN$=IN$+Z$:PRINT Z$;ZD$;Z$;
60110 IF Z=13 THEN PRINT CR$;:RETURN
60120 IF Z=20 AND ZL>0 THEN IN$=LEFT$(IN
$,ZL-1):PRINT"█";:GOTO 60010
60130 IF Z=141 THEN Z$=CHR$(-20*(ZL>0)):
FOR Z=1 TO ZL:PRINT Z$;:NEXT:GOTO 60000
60140 GOTO 60010
60200 ZJ=TI+30:ZT=TI:ZS=2:PRINT LEFT$(JC
$,1);PR$;"▔"
60210 IF(PEEK(JS) AND 16)=0 THEN ZS=2:GO
SUB 60280:PRINT:RETURN
60215 GOSUB 60500
60220 IF TI>=ZT THEN GOSUB 60280:ZT=TI+1
5:ZS=3-ZS
60230 Z=PEEK(JS)AND 12:IF Z=12 THEN ZJ=0
:GOTO 60210
60240 IF TI<ZJ THEN 60210
60250 IF Z=4 AND IN<JM THEN IN=IN+1:GOTO
 60200
60260 IF Z=8 AND IN>1 THEN IN=IN-1:GOTO
60200
60270 GOTO 60210
60280 PRINT TAB(JT+JW*(IN-1));MID$(JC$,Z
S,1);MID$(PR$,JT+JW*(IN-1)+1,JW);"▔"
60290 RETURN
60500 GET Z$:IF Z$<>"Q"THEN RETURN
60600 GET Z$:IF Z$<>""THEN 60600
60605 POKE VIC+24,21:POKE VIC+21,0:PRINT
 CHR$(9);
60610 GOSUB 61000:POKE VIC+33,6:POKE VIC
+32,14:POKE SID+24,0:PRINT"▔▔":END
61000 CRT=1024:VIC=53248:WD=40:CR$=CHR$(
13):SID=54272:JS=56320:CM=55296
61010 JC$="▔█":QL=214:QI=255:RETURN
62000 L0=LEN(PG$)+2:L1=LEN(AU$)+2:L2=LEN
(A2$)+2:IF L1<L2 THEN L1=L2
62010 IF L0<L1+2 THEN L0=L1+2
62020 B0$=LEFT$("██
                    ",L0+2)
62030 DEF FN T(N)=(40-N)/2:B1$="█"+MID$(
B0$,2,L1+1):T1=FN T(L1):T0=FN T(L0)
62035 GOSUB 61000:POKE VIC+32,0:POKE VIC
+33,0:POKE SID+24,0
62040 PRINT"▔███";CHR$(8):FOR I=1 TO 4:P
RINT TAB(T0);B0$:NEXT I
62050 PRINT"▔▔█";TAB(FN T(LEN(PG$)));PG
$:PRINT
62060 FOR I=1 TO 3:PRINT TAB(T1);B1$:NEX
T I
62070 PRINT"▔█";TAB(FN T(LEN(AU$)));AU$

62080 IF A2$<>""THEN PRINT"█";TAB(FN T(L
EN(A2$)));A2$:PRINT TAB(T1);B1$
62090 BG$="PULSE "+BG$+" P/ENTRAR":T0=FN
 T(LEN(BG$)+2)
62100 LN$=LEFT$("──────────────────────
──────────────",LEN(BG$))
62100 LN$=LEFT$("──────────────────────
──────────────",LEN(BG$))
62110 PRINT TAB(T0);"█████▔█r";LN$;"¬██
"
62120 PRINT TAB(T0);"L";LN$;"J"
62140 PRINT TAB(3);"█████COPYRIGHT (C) 1
985 THE CODE WORKS."
62150 L1=LEN(BG$):I=1:PRINT"▔▔▔▔▔▔▔▔"
62160 PRINT SPC(T0+1);MID$("█▶",I,1);LEF
T$(BG$,L1);"▔"
62170 L1=L1+1:IF L1>LEN(BG$)THEN L1=1:I=
3-I
62180 GET T$:IF T$<>""THEN 62200
62190 IF PEEK(JS)AND 16 THEN 62160
62200 CLR:GOSUB 61000:PRINT"▔█":GOTO 100
```

D E B U G G I N G

SIMULADOR DE VUELO

Les escribo para comentar mi experiencia en pasar el programa SIMULADOR DE VUELO del Nº 3 a un cassette, sabiendo que así colaboro con una revista que valoro mucho y con los lectores que, como yo, deben haber sufrido bastante a la par que pensábamos mal de ustedes injustamente.

Los problemas con que me encontré fueron de 3 tipos: 1) tenía los ingredientes de la torta y no sabía cómo mezclarlos ni cocinarlos, siendo mi primer experiencia en lenguaje máquina; 2) errores impresos en el programa generador de la línea REM con 2926 caracteres, además de una tijera mal implementada en el programa en BASIC; 3) errores de mi propia cosecha en el tecleado, algunos por simple torpeza y otros por no haber distinguido bien ciertos detalles.

Mi esfuerzo terminó bien y ahora disfruto sin inconvenientes el programa, por lo que quiero compartirlo.

Me aboqué a generar la línea REM y descubrí que en ese programa debería decir 2 LET A = PEEK 16396 + 256 * PEEK 16396-2, en cambio, donde está subrayado dice 255. Pude darme cuenta comparándolo con el gráfico de cómo queda la línea REM con 2926 caracteres (que en realidad corresponde a uno con 3000). También es diferente la línea 3 de como está corregida en el Debugging del Nº 6, porque el espacio en blanco está al principio y no al final (no sé si esto modifica el resultado). En dicho Debugging vuelven a equivocarse al intentar aclarar la línea 11, repitiendo 1,61 donde debe decir I,61.

Una vez que se tiene la línea 1 REM se agregar las líneas del cargador en lenguaje máquina, cuyo listado empieza con la posición 16514 y sigue con la 16515 inmediatamente al lado, cada renglón tiene 8 números que corresponden a 8 posiciones distintas y consecutivas (lo aclaro porque yo lo miraba sin entenderlo). En la pantalla sólo se podrá ver la línea 1 REM hasta el renglón 19 que termina con varios "?". Luego se agrega, empezando por la línea 2 el programa en BASIC. En este último, en la línea 8010, la tijera se comió renglones de espacio en blanco, siendo el objetivo formar 8 líneas en blanco.

Respecto a las cosas donde me equivoqué, las cuento porque pueden ser errores comunes a otros lectores. En la línea 7090 y 7091 se debe formar un cuadrado de 1 1/2 x 1 1/2 espacios, y no de 1 1/2 x 1. Además en las posiciones para pokear 17530 y 19178 van 33 y 203 respectivamente. En



la línea 9074 del BASIC la variable deberá valer −8 (en mi ejemplar está borroso).

Finalizando, para controlar los pokeados conviene agregar estas líneas:

```
9901 INPUT B
9902 FOR A=B TO 19440
STEP 8
9903 SCROLL
9904 FAST
9905 PRINT PEEK A;
TAB 4; PEEK (A+1); TAB
8; PEEK (A+2); TAB 12;
PEEK (A+3); TAB 16;
PEEK (A+4); TAB 20;
PEEK (A+5); TAB 24;
PEEK (A+6); TAB 28;
PEEK (A+7)
9906 SLOW
9907 IF INKEY$ = ""
THEN GOTO 9907
9908 NEXT A
9909 STOP
```

donde b será la primer posición a peekear. Para que coincida con el list... do conviene introducir a... guna de las mencionad... en la 1er. columna, p... diendo detenerse el pr... grama con BREAK, co... tinuar con GOTO 9903... con GOTO 9901 come... zar y recomenzar. La ... nea 9907 es para reg... lar el tiempo de aparici... de los datos.

Dos detalles, la rutin... "CRASH" comienza e... 19369 y no en 19364, el otro es que sería co... veniente, para mejor e... tender los gráficos qu... incluyan una plantilla tran... parente cuadriculada.

Suerte a Uds. con la revi... ta que es magnífica, y los lectores con éste ... otros programas.

M. Lew...

LA COMPUTADOR... VE POR NOSOTRO...

En el número 11, de f... brero pasado, en la pág... na 39, listado en códig... de máquina que es c... mún a todos los progr... mas de la serie prese... tada por Enrique Gonz... lez, le faltan estos byte... que se "cayeron" del p... de la página. Estos com... pletan un total de 218... bytes, o sea, de la dire... ción 63000 hasta 65180 inclusive.

1	37	2	3	2	34	12	10
1	14	1	5	1	5	0	11
1	4	2	3	1	14	0	0
255	0	255	0	255	0	255	0
255	0	255	0	255	0	255	10
200	100	100	10	200	100	100	10
200	100	100	0	0	0	0	0

EL MERCADO ARGENTINO DE LAS HOME COMPUTERS

Continuamos con nuestra investigación sobre los productos disponibles en nuestro país y sobre las aplicaciones que le dan los usuarios.



"Como una herramienta"
Ing. Jorge KATZ (OMIS)

La home computer es una máquina muy versátil. No es necesariamente, como está implantada en la Argentina, para jugar. Es mucho más potente y se le puede sacar mucho más provecho. Esta fue una de las causas por la que nos abocamos a hacer un tipo de programas que la implemente como una herramienta en una pequeña o mediana empresa.

Actualmente, el concepto de home computer, de parte de la gente que está interesada en computación, es el de un aparato para jugar. La gente va a comprar el "juguete" para los chicos, y el padre dice "No... yo no sé nada. No me explique nada porque el chico es el que sabe". Entonces se le vende el teclado, la unidad de almacenamiento, una diskettera o grabador, utiliza su televisor y el joystick.

Eso forma una unidad de venta que los chicos usan para jugar. Los juegos los consiguen en muchos negocios o por intercambio. Y así se pasan las horas.

Considero que no es la finalidad última de la máquina. Si bien puede resultar un entretenimiento en determinadas circunstancias, el objetivo principal de la máquina es poderle sacar provecho. Utilizarla como una herramienta. La computación no es un fin, es un medio. Es como una máquina de escribir o de calcular, que le permite a la persona hacer sus tareas en forma mucho más eficiente, con menos trabajo y mejores resultados y productividad.

¿Qué significa utilizarla como herramienta? Si un señor tiene una ciendo a mano el listado de recibo de jornales de los empleados, la máquina, preparando un programa de sueldos y jornales, le permite hacer lo mismo automáticamente.

La ventaja de las home computer con respecto a las PC es el costo. Mucha gente al no conocer la computación no se anima a hacer una inversión de varios miles de australes, y las home computer les permiten hacer el ensayo con mil o mil quinientos australes.

El usuario no está todavía muy convencido de lo que puede resultar pero lo intenta.

El lanzamiento de nuestros sistemas provocó ciertas anécdotas. Por ejemplo, gente que nos decía "no, eso no sirve" o "eso no se puede hacer", o "es imposible que con esa máquina usted pueda realizar lo que promete". Pero lo cierto es que hemos desarrollado sistemas que tienen muy buena acogida, y a la gente que realmente quiere hacer el cambio le resulta de mucha utilidad.

Desde hace un año está apareciendo una nueva mentalidad en el usuario. El hecho de usar la computadora permite aplicarla a cualquier actividad. Desde la actividad comercial e industrial hasta la médica, de archivo de datos, de mailing o procesadora de palabras.

Nosotros estamos en vías de establecer relaciones para la exportación de soft a los países periféricos. Hoy día, la Cámara Argentina de Software está realizando exportaciones con muy buenos resultados, y las perspectivas son muy alentadoras al respecto.

Cuando el público vea la diferencia de lo que es trabajar en las actuales circunstancias, y con un poquito de automatización, como es con la home computer, creo que se abrirá un mercado no tanto comercial como tecnológico que permitirá insertarnos en la computación actual.

En los próximos años todo el mundo va a estar con su pequeña computadora por lo que los costos seguirán bajando. Además, los lenguajes se van simplificando igual que los problemas.

Mucha gente habla de computación y dice "compro la máquina y voy a hacer un curso de con-

los manuales son tan explícitos que siguiéndoles paso a paso se puede llegar a utilizar la máquina. No se necesita ser programador porque los programas ya están hechos. Todos utilizan una calculadora, pero nadie se pone a estudiar cómo funciona la calculadora. Simplemente hay un librito que indica cómo se deben hacer las operaciones.

"Suplen un vacío"
Héctor Fernández (Pari)

Por supuesto que la actividad de la computación se introduce en las casas a través de los juegos. Luego, cuando el usuario que ya tiene una computadora, descubre que puede usarla en otra cosa, le empieza a sacar las máximas posibilidades. De acuerdo a la profesión del usuario la utilizará para su disciplina. El contador la va a usar para su trabajo, el médico para las historias clínicas, el arquitecto para los diseños y cálculos de estructura y graficación.

Actualmente diversos industriales y comerciantes la están empleando para sus especialidades. Nosotros hemos vendido home computers a estudios contables, estaciones de servicio, farmacias.

Las computadoras con capacidad profesional o semiprofesional suplen un vacío en la Argentina, porque si esa gente adquiría un equipo PC iba a tener el 60 u 80% de capacidad ociosa por el gasto que había realizado.

La home computer le permite realizar a la pequeña y mediana empresa todo aquello que le está vedado por el alto costo de un equipo PC y sobre todo ven una herramienta más para hacer más eficiente su gestión.

Administración es el manejo de los medios de información, y cuanta más información tenga para y de su comercio, más eficiente será. Por ejemplo, las industrias textiles que están utilizando la C-64 hacen un control de stock referido a los colores que se deben fabricar de acuerdo a todo lo vendido, utilizando una consola, una diskettera e impresora de carro ancho. También se utiliza la computadora en forma personal para después de integrar la información a un proceso industrial o comercial mayor.

En el área de la educación, una vez que se decanta la fiebre inicial por los juegos, que indudablemente son excepcionales e iguales a los de video, el padre y mismo el alumno, si va a un colegio que tenga gabinete de computación, se va formando en la disciplina que eligió con mayor eficiencia si utiliza la computación. Lo importante es acercarle el medio a la gente. Algunos lo van a usar, otros no, Pero aquellos que lo hagan, van a obtener un rendimiento más que proporcional que no lo hubieran obtenido sin la computación.

En cuanto a cómo está el mercado, creo que Argentina por su tradicional formación técnica de algunos de sus colegios, está igual de capacitada que muchos países de Latinoamérica como Brasil y México. Con un poco de esfuerzo y persistencia dentro de esta activi-



dad que recién se inicia, el país puede ubicarse en los primeros lugares como formadores de inteligencia y exportadores de productos. Situación que nos puede llevar a obtener divisas para el crecimiento.
Lo veo en los consumidores de computación que van de los 5 a los 18 años, que es la edad en que los chicos se forman. Hay un gran material receptivo de todo lo que sea inteligencia y computación, y como en el resto del mundo, todo ésto se está desarrollando.
Habría que hacer una conjunción entre obtención de divisas vía exportaciones agrícolas y obtención de divisas a través de una actividad que puede hacer que los muchachos se queden trabajando en la Argentina, obteniendo buenas remuneraciones y a la vez exportando. Porque el soft en castellano se puede exportar a Latinoamérica, ya que hay un mercado hispano parlante muy amplio. Pero para eso hay que tener la vocación de que queremos ganar dinero, que se quiera obtener divisas a través de una legislación de fondo que dé apoyo y defienda la propiedad intelectual.
Creo que a través de la resolución 44 puede darse que también se consigan buenos productos exportables en la medida que se combinen con inversiones extranjeras, que unido a lo que podemos llegar a dar de facilidades más la tradicional mano de obra calificada que necesita el hardware y el soft, con los colegios que tenemos, y un correcto apoyo en las universidades; podemos obtener algo que luego se podría trasladar a un desarrollo de la industria privada (que es en definitiva la que consigue mejores salarios). Y también le brindaría un apoyo a toda la estructura estatal que aparentemente es bastante ineficiente. La computación se puede trasladar a los distintos organismos oficiales, y todo integrado provocaría mayor celeridad. Las horas libres que le quedaría a la gente se podría aplicar a proyectos que ahora a lo mejor no se hacen producto de que se están llenando papeles o haciendo cosas improductivas.

Omis Informática

Distribuye y vende sistemas de gestión para la C-64 desarrollados por Sistemas Administrativos Modernos S.A. La configuración básica requiere una microcomputadora C-64, una unidad de diskette de 5 1/4, una impresora de 80 columnas y un monitor de video.
Ofrece el **Sistema de Sueldos y Jornales** (SAM-SYJ) que permite llevar la Administración de Personal de una empresa en forma simple y confiable. Sus características principales son las siguientes: amplia flexibilidad para el usuario en el armado de tablas y parámetros del sistema; los campos son denominados en el lenguaje común al área personal; mantenimiento de parámetros, de legajos, de familiares, de conceptos, por categorías; liquidación de un solo legajo, de complementarias, generales, de aguinaldo; aumento de categorías; cálculo de aportes y retenciones; emisión de recibo, libro ley, listado de cambio, totales por código; liquidación de haberes separados por jornalizados, mensualizados, y tres tipos a definir en parámetros para el usuario; mantenimientos de conceptos a liquidar.
Con esta configuración básica se puede manejar hasta 50 legajos y 99 conceptos, los cuales pueden ser depurados parcialmente por el usuario, posibilitando nuevos ingresos.
Además ofrece el **Sistema de Facturación y Stock** (SAM-FAC) que es un sistema totalmente interactivo que permite a una mediana empresa obtener información confiable, planificación y control. Se puede manejar hasta 600 artículos y 250 clientes.
El **Sistema de Cuentas Corrientes** (SAM-CTA) posibilita llevar las cuentas corrientes en forma rápida y segura facilitando las consultas selectivas por pantalla o impresora. En cada ingreso de Operaciones se informa el saldo del cliente. Se pueden manejar hasta 250 clientes y 1000 operaciones, las cuales pueden ser depuradas por el usuario permitiendo nuevos ingresos.
El **Sistema de Contabilidad General** (SAM-CON) lleva la contabilidad de una empresa pudiendo manejar hasta 350 cuentas y 1300 movimientos, los cuales pueden ser depurados parcialmente por el usuario posibilitando nuevos ingresos.
El **Sistema de Bancos** (SAM-BAN) lleva la cuenta corriente bancaria y la cartera de cheques posdatados de una empresa. Se pueden manejar hasta 100 bancos, 10 códigos de operaciones y 600 operaciones.
Todos los sistemas están diseñados basándose en equipos de mayor envergadura lo que posibilita futuras ampliaciones, de ser necesario, ampliando la configuración básica.

Distribuidora Pari

Trabaja con la línea de Drean Commodore, fundamentalmente la C-64 y la 128. En relación al soft ofrece: 1) Programas en castellano de **Sueldos** que emiten recibos, mantienen legajos de personal y liquidan retenciones y aportes patronales con emisión de listado, de **Facturación** en australes, y que trabajan en combinación con el de **Lista de Precios** que también está en australes. 2) Diversos utilitarios con manuales en castellano. 3) En breve lanzará un **Inventario** para 1200 artículos también en castellano.
Distribuye el lápiz óptico "Magic Pencil" con software registrado ley 11723/85.
Posee las siguientes características que lo diferencian de lo existente en el mercado: trabaja con efecto spray, se pueden confeccionar texturas propias de fondo de los dibujos a realizar (tiene 36 texturas programadas), trabaja con Fast-Load, tiene cable retráctil, viene con aproximadamente 24 dibujos de muestra de posibilidades de realización, y tiene salida para distintas impresoras.
Además dicta cursos de Logo, Basic, Assembler para C-64, 128 y Z80. También desarrolla cursos especiales para niños hipoacúsicos, dictados por docentes especializados, haciendo realidad las posibilidades de la computación en los diferentes campos de la educación y la salud.
En relación a las posibilidades que ofrece la C-64 y sus periféricos, pone en relieve que los empleados de fabricantes de centrales telefónicas por programa almacenado utilizan la C-64 como apoyo para el diseño de circuitos. También hay programas que convierten a la Commodore en una mini imprenta. Con papel común, en una Impresora 803, se puede confeccionar la papelería propia con impresión de membrete y publicidad. K64



GUIA PRACTICA

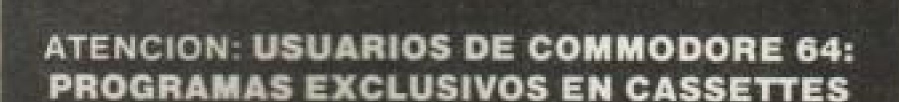


Conversión de T.V. Color y Atari
Service de Periféricos



Micro Electronic's
Av. DEL LIBERTADOR 3994 - T.E. 791-8316/797-7740

DISTRIBUIDOR OFICIAL
Drean commodore Le ofrece su
C 16 y C 64
• Sistemas de Computación
• Software (juegos y utilitarios)
• Accesorios
• Mesas de Computación
• Bibliografía
CURSOS: Basic y Atelier de Logo



Venta Por Mayor y Menor de: Interfases - Reset
Fundas - Transformadores - Reparación de
Consolas y Dattassete
Manuales en Castellano.

SAGO OMEGA
SANABRIA 3208 (1417)
TE.: 632-3191
SABADOS ABIERTO TODO EL DIA
ENVIOS AL INTERIOR

Para su Czerweny ahora si "Joystick"

Conecte su joystick directamente a su computadora CZ-1000, 1500, Spectrum y disfrute ya de su juego preferido

Adaptación + 1 joystick CZ-800 ₳ 19,.5
CZ-2000 SPECTRUM + 1 joystick + adaptación ₳ 259
CZ-1500 + 1 joystick + adaptación ₳ 148

INTELEC S.R.L.

Precio especial a distribuidores, adaptación autorizada por Czerweny
"conserve su Garantía"
Llámenos o consulte a su distribuidor.
Paraná 426 2do. Cuerpo Of. 1 Cap. 40-7000

Drean commodore

COMMODORE 64 - COMMODORE 16
PROGRAMAS - JUEGOS Y UTILITARIOS CON MANUALES
DISKETTERAS - DATTASETTES - FUNDAS
PROGRAMAS PARA C-128 - JOYSTICK
FAST LOAD C/RESET - DISKETTES INTERFACES

PEEK & POKE
Le ofrece además cursos de BASIC para Commodore c/manejo de archivos.

INFORMES: 784-7761 CAP. FED.
VIRREY ARREDONDO 2353 (altura Av. Cabildo 1500)

DISKETTES ¡No Camine más!

3,5" - 5 1/4" y 8" - Todas las marcas - Todos los modelos
ENTREGAMOS A DOMICILIO
CASAS PORTA DISKETTES - MEDIOS MAGNETICOS - FORMULARIOS CONTINUOS - CINTAS IMPRESORAS - MUEBLES P/COMPUTADORAS - ETIQUETAS AUTOADHESIVAS P/MAILING - ETC.
ENVIOS AL INTERIOR

ESTUDIO 2000
Av. Escalabarri Ortiz 2416 P.B. 4
L. a V. de 9 a 19 - 72-9887

"LIDER EN PRECIOS DE INSUMOS Y ACCESORIOS PARA COMPUTACION"

COMMODORE 64 - 128



COMPUTACION

SOFTWARE A MEDIDA
JUEGOS PARA CASSETTES Y DISKETTES
MANUALES EN CASTELLANO
JOYSTICKS - FUNDAS - ACCESORIOS

COMPRA, VENTA Y SERVICE

CIUDAD DE LA PAZ 2323 CAP. FED.
T.E. 784-0792

— SERVICE INTEGRAL —
SINCLAIR - COMMODORE
REFORMAS A PAL-N C 64/128
FUENTES C 64 ₳ 18

LOGICAL LINE

URUGUAY 385 OF. 404 T.E.: 45-2688/5020
46-7915 INT. 404

CASSETTE VIRGEN
PARA COMPUTACION
• Fabricación propia
• Utilizamos cintas Ampex Ferrocobalto
• Las medidas se preparan en el dia



Producciones ECCOSOUND S.A.
Tronador 611 - (1027) Cap.
551-9489 / 553-5080 / 553-5063

OFRECEMOS CALIDAD Y PRECIO
AL SERVICIO DE LA TECNOLOGIA
• CONSULTENOS • HAGA SU PEDIDO



NOVEDADES NOVEDADES NOVEDADES



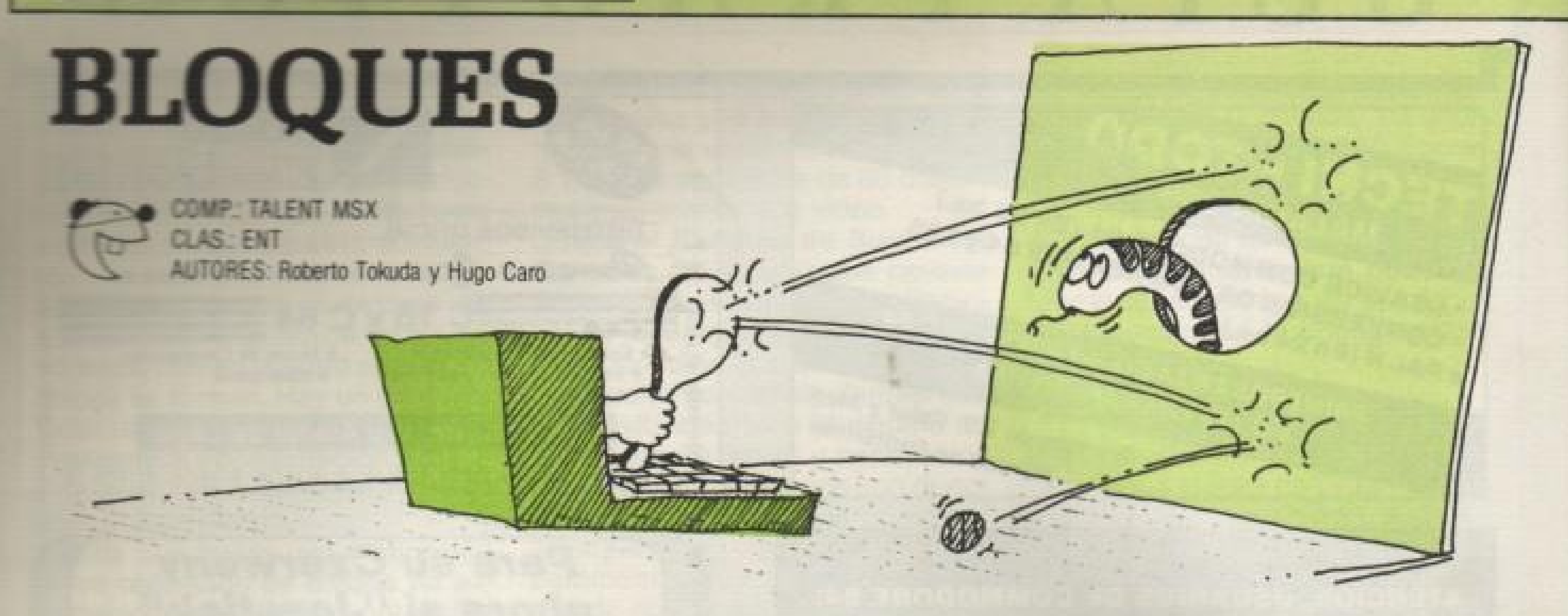
COMMODORE 64
ENVIOS AL INTERIOR SIN CARGO

MAS DE 2000 TITULOS EN JUEGOS, UTILITARIOS.
LO ULTIMO EN COPIADORES. TODOS LOS MANUALES
INGLES y CASTELLANO. EDUCATIVOS, CLASES DE INGLES, ETC.

ESMERALDA 740 - 5º OF. 512 (1007) 393-1608

BLOQUES

COMP.: TALENT MSX
CLAS.: ENT
AUTORES: Roberto Tokuda y Hugo Caro



El siguiente juego es una combinación de 3 juegos clásicos: El demoledor, el frontón y la víbora que evita ser tocada por algo.
En este caso, se trata de evitar que la pelota que rebota de un borde a otro de la pantalla toque a la calavera. Asimismo, si la pelota toca la cabeza de Bloques (la víbora), también se pierde. Si la cabeza toca la calavera también se pierde.
Para obtener puntos, la pelota debe ir tocando el cuerpo de BLOQUES.
El programa comienza preguntando si utiliza joystick o no. Contestemos "sí" en caso de poseer uno y conectémoslo en el port nº 1. Luego de esto comienza el juego. Para comandar a BLOQUES utilicemos los cursores (o las direcciones del joystick). Si finaliza el juego, pulsemos el disparador (o la espaciadora)

Para jugar nuevamente, o las teclas CTRL+ STOP para finalizar.
Es de notar la gran velocidad de reacción del MSX BASIC, ya que la única rutina de código de máquina que se utiliza es para imprimir en la pantalla de alta resolución con gran velocidad. K64

```
10 '
20 ' BLOQUES
30 ' DE ROBERTO TOKUDA
40 ' Y HUGO CARO
50 ' PARA REVISTA K-64
60 '
70 '
80 MAXFILES=1:CLEAR 100.&HD000:DEFUSR=&HD000:DEFUSR1=&HD010:
   DEFUSR2=&HD021
90 DEFINT A-Z:OPEN"GRP:"FOR OUTPUT AS #1
100 SCREEN 1:CLS
110 ' CARGA CODIGO MAQUINA E INICIALIZA
120 PRINT:PRINT
130 RESTORE 1430
140 AD=&HD000
150 READ D$:IF D$="FIN"THEN GOTO 170
160 D=VAL("&H"+D$):POKE AD,D:AD=AD+1:GOTO 150
170 INPUT"Utiliza Joystick?(s/n)";I$
180 J=0:TX$="Espaciador":IF I$="S"OR I$="s"THEN J=1:TX$=
    "Disparador"
190 XP=0:YP=0
200 XV=0:YV=0
210 TP=0
220 R=RND(-TIME)
230 SCREEN 2,0,0:KEYOFF
240 ON SPRITE GOSUB 1110
250 PD=4:HIT=0:BL=1:U=USR(0)
260 '
270 ' CARGA SPRITES
```

```
280 '
290 RESTORE
300 SCREEN 2,0
310 FOR K=0 TO 2:SP$="":FOR I=0 TO 7
320 READ R
330 SP$=SP$+CHR$(R)
340 SPRITE$(K)=SP$
350 NEXT:NEXT
360 '
370 ' PANTALLA PRINCIPAL
380 '
390 COLOR 15,1,1:CLS
400 LINE(63,0)-(242,186),15,B
410 PRESET(8,24):PRINT#1,"BLOQUES"
420 PRESET(8,56):PRINT#1,"PUNTOS"
430 PRESET(8,64):PRINT#1,STRING$(6,&H9D):U=USR2(0)
440 PRESET(8,96):PRINT#1,"TOPE"
450 TP$=RIGHT$("00000"+RIGHT$(STR$(TP),LEN(STR$(TP))-1),6)
460 PRESET(8,104):PRINT#1,USING"&    &";TP$
470 FOR I=20 TO 60 STEP 8
480 XP=65:YP=2:XS=6:YS=6
490 GOSUB 1030
500 XB=100:YB=100
510 PUT SPRITE 1,(XP,YP-2),12,1
520 I=BL:FOR BL=1 TO I
530 GOSUB 1060:NEXT:BL=I
540 '
550 ' PAUSA PARA INICIO
560 '
570 PSET (78,100),0:PRINT #1,"Pulse el "TX$
580 IF STRIG(J)=0 THEN 580
590 LINE (78,100)-(240,108),1,BF
600 BEEP:SPRITE ON
610 '
620 ' RUTINA PRINCIPAL
630 '
640 IF 3=STICK(J) AND 227>XP THEN XP=8+XP:GOTO 680
650 IF 7=STICK(J) AND 71<XP THEN XP=XP-8:GOTO 680
660 IF 1=STICK(J) AND 7<YP THEN YP=YP-8:GOTO 680
670 IF 5=STICK(J) AND 176>YP THEN YP=YP+8 ELSE 720
680 PUT SPRITE1,(XP,YP-2),15,1
690 '
700 ' DIBUJA BLOQUES
710 '
720 LINE (XP,YP)-(6+XP,5+YP),8,BF
730 '
740 ' CHEQUEA REBOTE PELOTA
750 '
760 XB=XB+XS:YB=YB+YS
770 IF 231<XB THEN XS=-ABS(XS):PLAY"L64A":GOTO 790
780 IF 64>XB THEN XS=ABS(XS):PLAY"L64A"
790 IF 1>YB THEN YS=ABS(YS):PLAY"L64A"
800 IF 183<YB THEN YS=-ABS(YS):PLAY"L64A"
810 PUT SPRITE0,(XB,YB),12,0
820 '
830 ' ACERTADO REBOTE
```

```
840 '
850 IF 8<>POINT(3+XB,3+YB) GOTO 880
860 PLAY"L64B":PAINT (3+XB,3+YB),1:HIT=1+HIT:U=USR1(0)
870 YS=-YS
880 IF 160*BL=HIT THEN BL=1+BL:GOSUB 1060
890 GOTO 640
900 '
910 ' RUTINA CHOQUE Y FINAL
920 '
930 PLAY"L501A":PD=PD-1
940 FOR I=0 TO 1000:NEXT
950 IF PD<>0 THEN FOR I=0 TO 3000:NEXT:GOTO 290
960 IF 10*HIT>TP THEN TP=10*HIT
970 LINE(72,72)-(232,112),1,BF
980 PRESET (120,80):PRINT #1,"FIN JUEGO":PRESET(72,96):PRINT #1,
    "Otro juego:"TX$
990 IF STRIG(J)=-1 THEN 250 ELSE 990
1000 '
1010 ' DECREMENTA CONTADOR BLOQUES
1020 '
1030 LINE(47,168)-(8,160),0,BF
1040 PRINT #1,"BLQ";PD
1050 RETURN
1060 XG=66+INT(RND(1)*22)*8:YG=8+INT(RND(1)*22)*8
1070 PUT SPRITE BL+10,(XG,YG),4,2:RETURN
```

```
1080 '
1090 ' APAGA DETECCION COINCIDENCIAS
1100 '
1110 SPRITE OFF:BEEP
1120 RETURN 930
1130 '
1140 ' DATOS SPRITES
1150 '
1160 DATA &B00111100
1170 DATA &B01111110
1180 DATA &B11111111
1190 DATA &B11111101
1200 DATA &B11111111
1210 DATA &B11111001
1220 DATA &B01110010
1230 DATA &B00111100
1240 '
1250 DATA &B11111111
1260 DATA &B10000001
1270 DATA &B10000001
1280 DATA &B10000001
1290 DATA &B10000001
1300 DATA &B10000001
1310 DATA &B10000001
1320 DATA &B11111111
1330 '
1340 DATA &B01111100
1350 DATA &B11111110
1360 DATA &B10010010
1370 DATA &B10010010
1380 DATA &B11101110
1390 DATA &B01111100
1400 DATA &B01111100
1410 DATA &B01010100
1420 ' DATOS CODIGO DE MAQUINA
1430 DATA 21,51,D0,3E,06,36,00,23
1440 DATA 3D,20,FA,C9,00,FE,41,20
1450 DATA 21,52,D0,06,05,7E,3C,27
1460 DATA E6,0F,77,20,04,23,05,20
1470 DATA F4,21,51,D0,06,06,7E,87
1480 DATA 87,87,C5,E5,21,57,D0,16
1490 DATA 00,5F,19,E5,21,00,08,CB
1500 DATA 20,CB,20,CB,20,48,06,00
1510 DATA 09,54,5D,E1,01,08,00,CD
1520 DATA 5C,00,E1,C1,23,05,20,D6
1530 DATA C9,00,02,00,00,00,00,00
1540 DATA 3C,46,46,46,46,46,3C,00
1550 DATA 1C,3C,7C,1C,1C,7E,00
1560 DATA 3C,46,46,1C,30,60,7E,00
1570 DATA 3C,46,46,0C,46,46,3C,00
1580 DATA 06,0E,16,26,7E,06,06,00
1590 DATA 7C,60,60,7C,06,46,3C,00
1600 DATA 1C,20,40,7C,46,46,3C,00
1610 DATA 7E,46,46,0C,18,18,18,00
1620 DATA 3C,46,46,3C,46,46,3C,00
1630 DATA 3C,46,46,3E,06,06,3C,06
1640 DATA 00,FF,00,FF,00,FF,00,FIN
```

TRUCOS, TRAMPAS Y HALLAZGOS



Capacidades gráficas

Mediante estos dos cortos programas podemos ver como aprovechar las capacidades gráficas del ZX 81 y compatibles. En ambos casos debemos tener la máquina en SLOW.

Una vez que hayan visto los resultados notarán que con pocas líneas y mucha imaginación se puede más que al revés.



```
 20 LET A=INT (RND*42)+1
 30 LET B=INT (RND*42)+1
 40 LET C=INT (RND*2)+1
 50 FOR J=1 TO A STEP C
 60 PLOT A,J
 70 PLOT J,A
 80 PLOT B,J
 90 PLOT J,B
100 NEXT J
110 GOTO 20
 10 FOR N=40 TO 60
 20 FOR I=1 TO N
 30 LET R=I*22/N
 40 LET T=PI*100*I/N
 50 PLOT 32+R*COS T,22+R*SIN T
 60 NEXT I
 70 PAUSE 100
 80 CLS
 90 NEXT N
```

Listado particular

Para lograr un efecto visual en un listado, le sugerimos probar el siguiente programa:
10 REM YNC
Luego, tipee:
POKE 16513,56 y presione ENTER;
POKE 16517,147;
ENTER nuevamente.
La computadora debe estar en SLOW.
El problema aparece cuando queremos tomar control de la máquina nuevamente, **dejamos a nuestros lectores la resolución** del mismo.

Sutileza de FAST y SLOW

Una forma menos convencional para pasar al modo FAST, puede lograrse mediante un CALL a la ROM. Concretamente, si hacemos RANDOMIZE USR a la dirección 02E7 (ojo, está en hexadecimal) podremos asegurarnos trabajar en FAST, con la ventaja de que una posterior llamada a 0207 deja a la máquina en el estado en que estuviera antes, sea FAST o SLOW.



PREMIOS DEL MES

COMPUTADORAS - CASSETTES - BECAS

SUSCRIPTORES

GANADOR DEL MES, SORTEO CZ 1000

ALEJANDRO YADRONIK
Villa Celina - Pcia. Bs. As.

FELICITACIONES

DEBERAS RETIRARLA EN NUESTRAS OFICINAS CON LA PRESENTACION DEL D. DE IDENTIDAD

SORTEO ENCUESTA: GANADORES DEL MES

PREMIO: 10 BECAS

MARIA C.L. CALVINHO
LEANDRO D. BENMERGUI
OSCAR E. MACIA
FERNANDO LUCERO
JOSE A. BELMONTE
GABRIEL OSTA
GUSTAVO HARTINGH
CARLOS SCHRECK
FERNANDO SOTO
PABLO G. RIAL

PREMIO: 40 CASSETTES

SILVINA CASTRO KUBAT	Capital
GERMAN CAMUSSI	Córdoba
FERNANDO BONNIM	Don Torcuato
MARCELO DE NADAI	San Martín
CECILIA GARABUAU	Concordia
ADRIAN J. KHANIS	Capital
NILDA PIRES	Capital
CESAR O. VENA	Azul
PEDRO A. TESONE	Capital
ALBERTO IRUMBERRI	Rauch
JUAN F. QUIROGA	Córdoba
JUAN CARLOS CEPEDA	Las Heras
MAURO PETRILLO	Haedo
RAUL JIMENEZ	Capital
ADRIANA MENDIBERRI	Neuquen
RUBEN B. MANSUR	Resistencia
NESTOR A. FERRARA	Rio Negro
SANDRA SCIBONA	L. de Zamora
MARCELO A. CERNIATO	La Plata
MIGUEL A. MONACO	Solano
MARCELO R. BELVEDERE	Gral. Rodriguez
OSVALDO CASAIS	Villa Urquiza
JAVIER VAGLIENTE	Córdoba
LUCAS CABANILLAS	Galvez Santa Fe
JUAN A. CASALLA FERRANDO	La Plata
CLAUDIA S. ORTIZ	Mercedes
EMILIO ARAKAKI	R. de Escalada
FEDERICO PARODI	Victoria
SERGIO F. PERRIN	Punta Alta
MARIA S. TOSCANO	Azul
EDUARDO BERGUÑO	Ramos Mejia
LUCAS L. VILLAGRA	Tucuman
LEANDRO VESCO	Paraná
EDGAR A. BEA	San Luis
MARTIN GALICH	Marcos Juárez
ORLANDO D. CORONEL	Ciudadela
MIGUEL A. RIUS	Rawson
FERNANDO QUEVEDO	Villa Maria
JUAN A. APTER	Corrientes
EZEQUIEL CHESINI	Rio Gallegos
ESTEBAN ETCHEVERRY	La Plata

Los premios podrán retirarse en la Administración de K64, Cerrito 1320 - 1º P. Capital, con documento de identidad en el horario de 10 a 12 y 15 a 17 hs. Quienes viven en el interior del país, pueden solicitar que se les remitan los premios por correo.

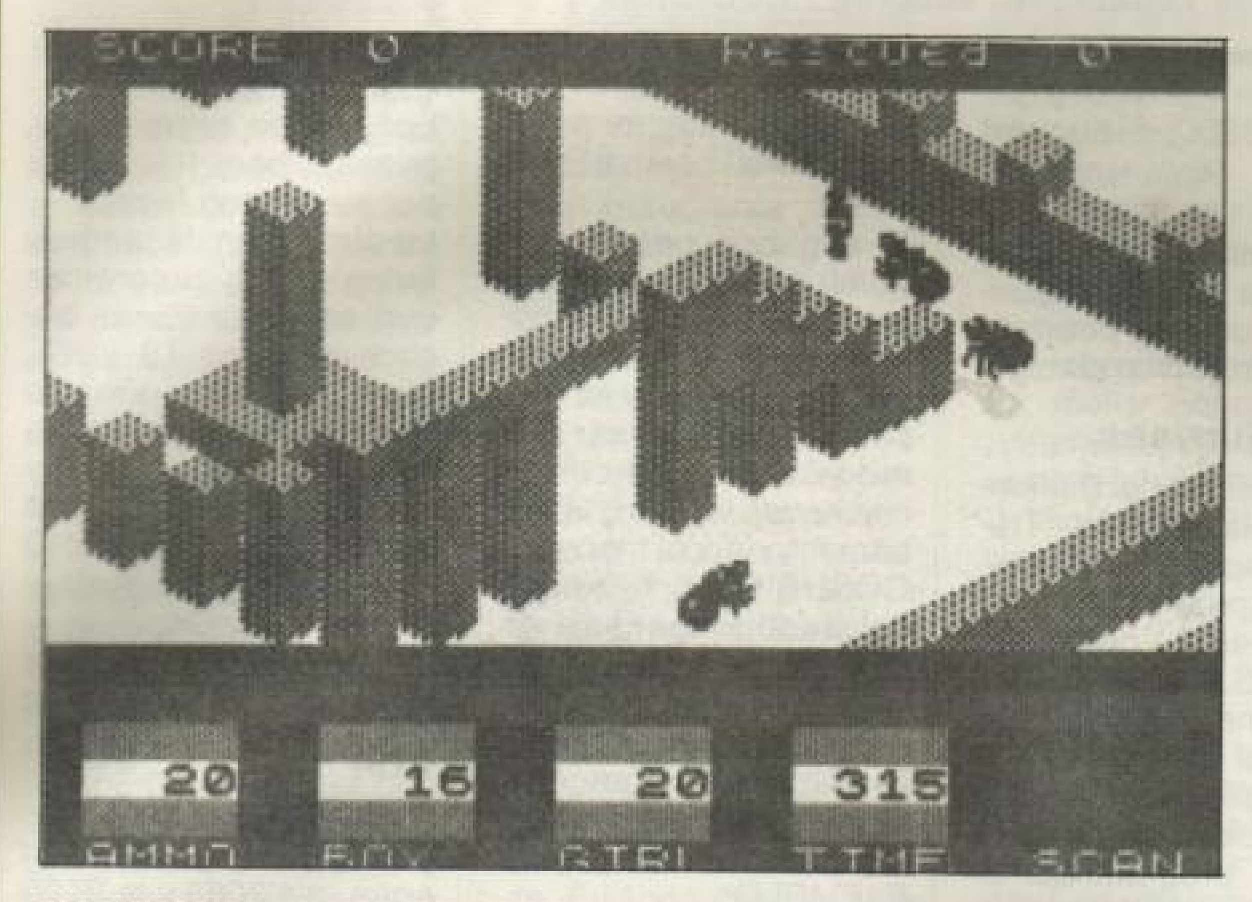
RAID OVER MOSCOW



CLASIFICACION: JUEGO DE ACCION
COMPUTADORA: COMMODORE 64
CONFIGURACION: 64 K
FACTOR K-64: 9
HECHIZO: 9
DOCUMENTACION: N/D

Esta vez nos toca estar al frente de una misión para detener y destruir los misiles nucleares que la Unión Soviética ha lanzado contra distintos puntos del mundo libre.
La acción de RAID OVER MOSCOW varía en su tipo y nivel de dificultad a medida que avanzamos en el juego. La acción comienza con una vista del hemisferio norte del planeta, mostrando la detección del lanzamiento de misiles de uno de cuatro silos situados en Rusia.
En el nivel de principiantes, tenemos sólo siete minutos para detener la guerra nuclear. Como piloto de caza que somos, debemos rápidamente despegar, ponernos en camino a Rusia y atacar los silos de misiles.
Tan solo para despegar se necesita una habilidad considerable, dado que debemos manejarnos en un ambiente de escasa gravedad y, maniobrar la nave en estas condiciones, no es cosa fácil.
Los controles son muy realistas. El secreto para poder despegar con éxito consiste en tomar control de la nave apenas ésta se levanta del suelo y al estar a la altura correcta abrir las puertas del hangar.
Se pueden volar hasta nueve aviones por juego y es una pena que al comienzo se pierdan la mayoría de ellos tratando de salir del hangar, siendo ésta parte la más aburrida del juego.
El territorio enemigo está defendido por misiles guiados por calor que se aproximan por la retaguardia y numerosas armas de tierra. Se debe volar a baja altura para evitar los misiles, a la vez que se evitan los árboles y se destruyen las estaciones de tierra. La respuesta del avión es eficiente y el scroll del terreno esta muy bien logrado.
Destruir los silos de los misiles no es una empresa tan difícil. Estos están defendidos por cohetes y la aviación enemiga. Sólo el silo del medio es el importante, pero los de los costados nos dan más puntos y bonus. La pequeña ventana por la que disparan es su punto débil. Una vez que estamos alineados con la ventana, ésta se vuelve azul. Destruído un sitio de lanzamiento se pasa al próximo, hasta que se destruyan tres.
Luego pasamos al ataque del Centro de Defensa Soviético; es, sin dudas, una de las pantallas mas divertidas. Tenemos que abrirnos paso entre soldados enemigos y un tanque. Una vez hecho esto llegamos al cuarto del reactor. Esta es la parte final del juego y sólo les diremos que no será nada fácil salir con vida de este lugar.
Los gráficos del juego están muy bien definidos, y el sonido es sumamente realista. La nave responde a los controles en forma rápida y eficiente, y el juego tiene un nivel de dificultad realmente alto.
Pese a tener tres niveles de dificultad, incluso en el primero, se hace difícil llegar a las partes finales del juego (en especial las dos últimas) que son las más divertidas.
RAID OVER MOSCOW es un juego bien hecho, que no nos va aburrir por el juego en sí mismo, aunque nos pueda decepcionar por no poder lograr una buena preformance hasta haber pasado unas cuantas horas con él.

ANT ATTACK



CLASIFICACION: JUEGO DE ACCION
COMPUTADORA: SPECTRUM - TK-90X
CONFIGURACION: 48 K
FACTOR K 64: 7
HECHIZO: 7
DOCUMENTACION: N/D

Nos encontramos sin dudas frente a un clásico de la línea SPECTRUM Este juego tiene como principal característica que no desciende de ningún otro clásico (por ej. derribar invasores del espacio) y utiliza una nueva técnica tridimensional que su autor, Sandy White, conserva como su más preciado secreto.

El juego comienza con una presentación en donde se nos introduce en la misión que debemos llevar a cabo y, algo no muy común, somos requeridos acerca de nuestro sexo. Parece que por fin les dieron un lugar a las chicas en los juegos donde el heroe rescata a la doncella, porque aca la situación puede ser invertida y ser la doncella, porque aca la situación puede ser invertida y ser la doncella quien rescata al heroe.

Una vez dentro de la ciudad de las hormigas debemos rescatar a quien se halla prisionera y sacarla de la ciudad. Esto debe hacerse antes de que el reloj llegue a cero, en cuyo caso la aventura habrá concluído, y sin que las hormigas nos piquen más de veinte veces, tanto a nosotros como a la doncella que corre a nuestras espaldas.

La ciudad está muy bien realizada y aquí es donde se encuentra la mayor atracción del juego, en el efecto tridimensional. Es como si nosotros viéramos la acción a través de una de cuatro cámaras de televisión que se hallan en las esquinas de una habitación. Mediante cuatro teclas podemos seleccionar una de las cámaras y así poder apreciar la acción desde distintos angulos. De este modo nunca quedaremos ocultos por un edificio, basta con cambiar el ángulo de visión.

El control del movimiento debe hacerse mediante el teclado, dado que en su versión original no incluye la opción de utilizar joystick, Para matar a las hormigas disponemos de veinte bombas que pueden ser arrojadas más o menos lejos según la tecla que apretemos. De todos modos el funcionamiento de estas granadas no es del todo efectivo y cuando más las necesitamos misteriosamente no hacen blanco en las hormigas.

Los movimientos están muy bien logrados y el scroll de la pantalla, a medida que nos movemos por la ciudad, es suave y cómodo para la visión.

Los gráficos son buenos, no tienen gran colorido porque no es realmente necesario y el juego posee escasos sonidos.

El programa está grabado a velocidad normal, y no tuvimos problemas para su carga.

Ant Attack es un juego que tardara mucho en resultarnos aburridos y es uno de los clásicos que no deben faltar en una buena biblioteca de juegos.

En esta sección atendemos todas aquellas consultas y sugerencias que nuestros lectores deseen realizar. Para ello sólo debe dirigirse a esta redacción, sección "Consultas".

Intercambio TK 2000

Quiero comunicarme con todos aquellos que posean una TK 2000 y estén interesados en intercambio de software e información sobre dicho microcomputador.
Desgraciadamente en nuestro medio nadie se ha interesado en ocuparse del mismo, y toda la bibliografía y programas vienen del Brasil.
Por eso intento reunir a quienes se interesen, utilizando la revista de mayor difusión en Argentina.
Desde ya muchas gracias.

Hugo H. Campanelli
Suárez 1729 8 P. "C"
(1288) Capital

TI en Búsqueda:

Me gustaría comunicarme con Clubes de Usuarios de TI99/4A.

Adriana Mendiberri
Eduardo Talero 975
8300 - Neuquén

K64

Publicamos tu dirección completa con la seguridad de que se pondrán en comunicación contigo.

34 Columnas

En la revista de Diciembre, en la sección "Trucos Trampas y Hallazgos" dan un programita para escribir en 34 columnas con la TS 1000, pero borra las marcas de fin de línea y al hacer PRINT o CLS se pierde el control.
La solución está en estos dos programitas. El primero es similar al que publicaron ustedes pero deja el archivo de pantalla con sus marcas de fin de línea tal como estaba antes.
El segundo permite imprimir cualquier mensaje en la pantalla de 34 columnas. Está compuesto por 3 subrutinas y es muy sencillo; para pasar a la modalidad de 34 columnas hacer: GOSUB 9600; para imprimir el mensaje: GOSUB 9000, habiendo colocado el mensaje en la variable a$ de la siguiente forma: los dos primeros caracteres son el número de línea (1 a 22), los dos restantes son los de la columna (1 a 34)
EJEMPLO:
Para hacer PRINT AT 15, 7; "HOLA"; DEBE-MOS HACER: LET A$ = "1406 HOLA" Y GOSUB 9000
Para volver a modalidad normal: GOSUB 9500
También les ofrezco mi aporte a los crash de lujo; prueben con PRINT USR 2.

2068 ó TK 90X

Les mando estas líneas, primero, para felicitarlos por su exitosa revista.
La otra razón es preguntarles si los programas que son para tipear, hechos con una TS 2068, pueden tipearse con una TK 90X, ya que lo intente con el programa GATE y no pude hacerlo puesto que el paso 7130 SOUND 8,0,... no tiene equivalente en mi máquina.

Guillermo F. Rutman
Rosario

K64

Con respecto a su pregunta, la compatibilidad entre una 2068 y la TK 90 es la misma que entre la 2068 y la SPECTRUM. Por lo tanto no hay muchas sentencias que no

Listado 1

```
10 LET P=PEEK 16396+256*PEEK 16397+1
20 FOR A=1 TO 22
30 FOR B=1 TO 34
40 POKE P,1
50 LET P=P+1
60 NEXT B
70 NEXT A
100 IF INKEY$="" THEN GOTO 100
200 LET Z=PEEK 16396+256*PEEK 16397
210 FOR X=Z TO Z+726 STEP 33
220 POKE X,118
230 NEXT X
```

Listado 2

```
100 GOSUB 9600
120 LET A$="1001ESTA ES UNA LINEA DE 34 CARACTERES
130 GOSUB 9000
500 IF INKEY$="" THEN GOTO 500
510 GOSUB 9500
520 STOP
9000 REM IMPRIMIR EN 34 COL.
9010 LET X=Z+(VAL A$( TO 2)*34+VAL A$(3 TO 4))-35
9020 FOR A=1 TO LEN A$-4
9030 POKE X+A,CODE A$(A+4)
9050 RETURN
9500 REM VOLVER A LA NORMALIDAD
9510 FOR X=Z+33 TO Z+726 STEP 333
9520 POKE X,118
9530 NEXT X
9540 RETURN
9600 REM PREPARAR ARCH DE IMAGEN
9610 LET Z= PEEK 16396+256*PEEK 16397
9620 FOR X=Z+33 10 Z+726 STEP 33
9630 POKE X,0
9640 NEXT X
9650 RETURN
```

tengan equivalente, y una de ellas es el comando SOUND.
Si bien no es lo mismo, éste puede ser reemplazado con el comando SOUND de la TK 90, aunque su frecuencia y duración deberá ser probada hasta dar con sonidos de tu gusto. Lo que ocurre es que la TS2068 permite en ese comando, mayor control sobre el sonido generado; por eso tiene un argumento de más de dos números.

Sonido con TK 85

Ante todo quiero felicitarlos por la revista, está super buena.
Soy poseedor de una TK 85 y quisiera saber dos cosas:
1 - Me he enterado que existe un generador de sonido para la TK 85. ¿Cómo se usa y dónde se lo conecta a la computadora?
2 - Si ustedes o algún lector me pudiera enviar una copia del programa "Simulador de vuelo", ya que acá se agotó rápido la revista y no me fue posible adquirirla.
Desde ya muchas gracias.

Aníbal Adrián Londero
Don Bosco 1660
(3100) Paraná (E.R.)

K64

1 - El generador de sonido que conocemos se conecta al port de expansión trasero de la máquina y para su uso deben utilizar pokes adecuados, aunque existe software que facilita esta operación.
2 - Esperamos contestar a tu llamado.

TI vs. Commodore

Soy estudiante de segundo año y estoy próximo a comprarme un microcomputador TI 99/4A. Ustedes al responder a un lector le dijeron que era un modelo anticuado y que la más conveniente era la Commodore.
Yo, por un lado veo al sistema operativo de la TI 99 más "fácil", comparado al de COMMODORE. Quiero que me informen si estoy en buen camino y si para ustedes ser anticuado es ser ineficiente. En todo caso me podían decir los pro y los contra de estas dos máquinas.

Sergio Leguizamón
Wilde

K64

En primer lugar, el que te hayas decidido a comprar una microcomputadora ya es un buen camino.
En cuanto a qué máquina elegir, si la TI o la COMMODORE, no podemos decirte que compres una u otra, Esta decisión te toca tomarla a vos. Analizar los pro y contras de cada máquina nos llevaría hojas, pero podemos darte un par de bases sobre las cuales fundamentar tu decisión.
1 - Memoria libre para el usuario: la C-64 tiene 64 K Ram de fábrica, la TI tiene muchos menos. Averigua cuanto te costaría expandirla.
2 - Expansiones: ¿Cuánto cuesta ponerle dikettera a una u otra máquina? Aunque no la pienses comprar de inmediato, es un factor importante si pretendes darle algún uso serio a la computadora.
3 - Software: qué cantidad, variedad y precio. Compararlos.

TRUCOS Y TRAMPAS

Debo felicitarlos por la revista y decirles que las sección "Trucos y Hallazgos" es realmente interesante ya que poseo una CZ1500 y tengo un programa (Frogger) imposible de breakear y siguiendo sus consejos pude hacerlo y desarmarlo todo.
¿Qué dice en la línea 200 de Interceptor· Galáctico?

Eduardo Dorola
S.A. de Padua

K64

Gracias, y por lo de la la línea 200, ya salió en el número anterior la rectificación.

Zeus

Les escribo con motivo de hacerles la siguiente consulta: tengo una 2068 y me ha resultado de gran interés el artículo "Computadoras que hablan".
Con él se ofrece un programa en código máquina. Me gustaría saber cómo tipear este programa y si se necesita un ensamblador. En tal caso, podrían informarme ¿qué es, cómo es y cómo se utiliza un ensamblador?

José Luis Coccoz
Venado Tuerto - Santa Fe

K64

El programa a que hacés referencia debe ser cargado con un ensamblador. En cuanto al uso del mismo podrás verlo en forma detallada en la nota "CARGANDO COMO LOS DIOSES" donde se explica cómo utilizar el ZEUS ASSEMBLER.

Bolsa de Usados

Compro Impresora Omnis 850 para Texas TI 99/4A.

Héctor M. BEIM
Cerrito 159
ROSARIO
TE. 820186

Vendo TI 99/4A con 5 módulos (Basic extendido; Parsec; Microcirugía; Ajedrez; Home financial Dec) y 105 programas (incluye curso Basic); cable p/grabador; funda; manuales; etc. Todo original y en perfecto estado: ₳ 350.-

TE. 797-5334 - Sergio
Capital Federal

Vendo pantalla AMDEK 12" Video Monitor Model 310/310A W/CARD sin uso: ₳ 350.-

TE. 312-2141/8505 de Lun. a Vier. de 10 a 19 hs.

• VENDO TK-85 en perfecto estado, con joystick y 20 juegos. Todo A 140.- Christian Scivetti, Sarmiento 4995, esquina Río Gallegos, Ezpeleta, Pcia. de Bs. As.

• VENDO TK-83 en expansor, cables, fuente, manual y caja original, más 10 programas, todo como nuevo. Guillermo Olmos, San Martín 58, (7240) Lobos, Pcia. de Bs. As.

• VENDO interfase 1; microdrive; 4 cartuchos, software e impresora térmica Alphacom 32, Te.: 825-2302

• VENDO Cartridge emulador Spectrum para TS 2068; Te.: 825-2302

SORTEO-ENCUESTA K64

LLENE ESTE CUPON Y PARTICIPE DEL SORTEO MENSUAL



50 PREMIOS:
40 CASSETTES Y 10 BECAS PARA CURSOS BASIC

NOMBRE:... EDAD. ... OCUPACION:........................ TEL:........................

DIRECCION:.. C.P.:..... LOCALIDAD:........................ PCIA:........................

ENCUESTA

COMPUTADORA: ☐ CZ 1000 ☐ CZ 1500 ☐ CZ 2000 ☐ TIMEX 2068 ☐ TK 83 ☐ TK 85 ☐ TK 90
☐ C-16 ☐ C-64 ☐ TI 99/4A ☐ Otras .... ☐ NO TENGO AUN

ME GUSTARIA VER:

MAS	IGUAL	MENOS	
☐	☐	☐	PROGRAMAS DE APLICACION ESPECIFICA
☐	☐	☐	PROGRAMAS EN BASIC
☐	☐	☐	PROGRAMAS EN LOGO
☐	☐	☐	PROGRAMAS EN LENGUAJE DE MAQUINA
☐	☐	☐	PROGRAMAS EN OTROS LENGUAJES
☐	☐	☐	ANALISIS DETALLADOS DE LOS PROGRAMAS
☐	☐	☐	NOTAS PARA BEGGINERS

MAS	IGUAL	MENOS	
☐	☐	☐	JUEGOS
			CALIFICACION DESCRIPTIVA DE:
☐	☐	☐	PROGRAMAS DE JUEGO
☐	☐	☐	PROGRAMAS DE APLICACIONES COMERCIALES
☐	☐	☐	PROGRAMAS EDUCATIVOS
☐	☐	☐	HARDWARE

QUE ES LO QUE MAS TE GUSTA DE K64?

QUE ES LO QUE MENOS TE GUSTA?

Enviarlo a: K64 Computación Para Todos - Cerrito 1320 1° (1010) Buenos Aires, Rep. Argentina



Nº 3 ₳ 2.00

REP. ARGENTINA

Programas

DIRECTORIO SIMULTANEO
MEZCLADOR DE PALABRAS
CALCULOS MATEMATICOS

Simons' Basic:

PROGRAMACION ESTRUCTURADA

Assembler:

INCREMENTOS EN LOS REGISTROS

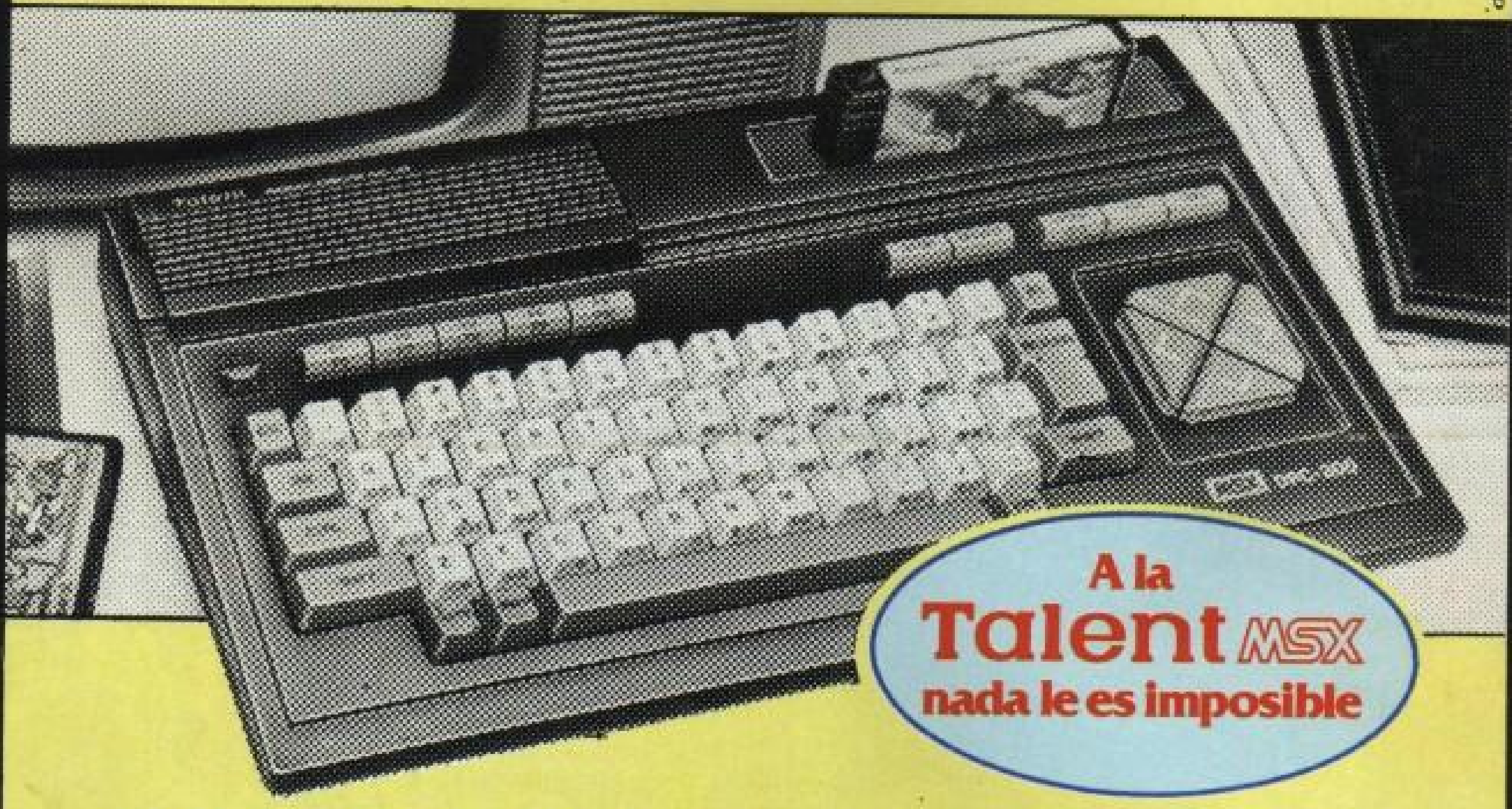
Direcciones de memoria

Drean Commodore 16:

GRAFICOS DE ALTA RESOLUCION

¡YA! CLUB DE USUARIOS Talent MSX Chile 1345

dialogo - D.P.



El Club de Usuarios de MSX ya funciona en Chile 1345

Invitamos a los felices usuarios de la TALENT MSX al curso gratuito de introducción al fabuloso mundo de MSX.

Participe del Club de Usuarios de MSX y encuéntrese con sus amigos que también tienen la TALENT MSX, e intercambiará programas, datos y chimentos.

Podrá probar todos los accesorios de la línea MSX, ¡¡desde disketteras hasta robots!!

Podrá ver y leer todo lo que le interese sobre la norma MSX: catálogos, libros y revistas de todo el mundo.
Todo con la seguridad, respaldo y seriedad que sólo TALENT puede brindarle.

¡Para inscribirse, no olvide traer su factura de compra!

Club Talent MSX

offset macchi